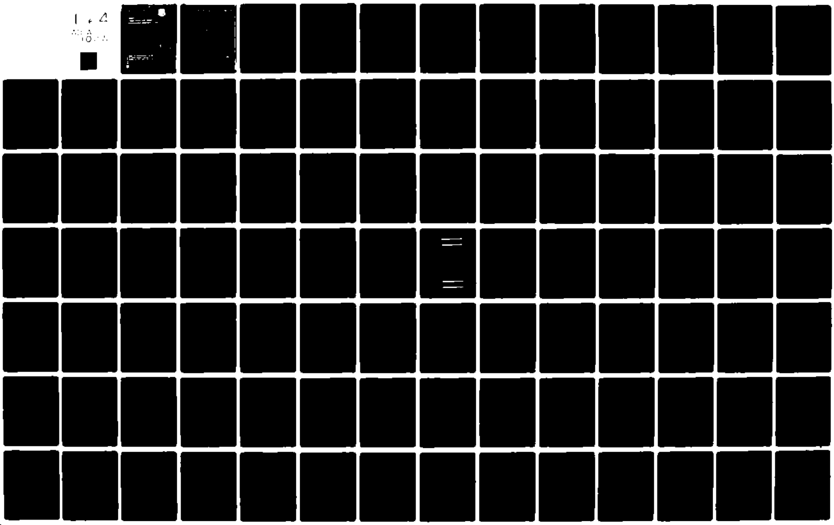
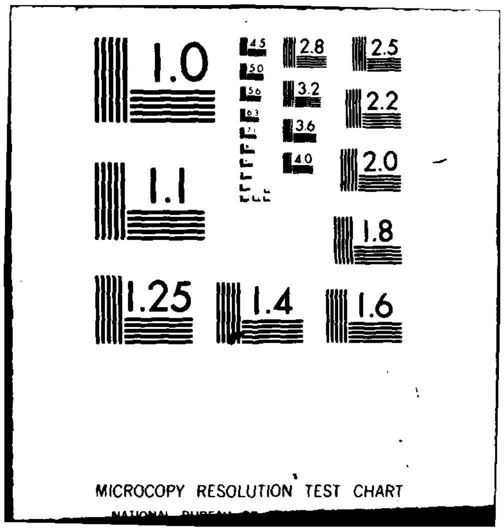
AD-A110 272 PAR TECHNOLOGY CORP ROME NY F/G 9/2
SABERS. STAND-ALONE ADIC BINARY EXPLOITATION RESOURCES SYSTEM. --ETC(U)
SEP 81 A J FRANKLIN, R L CALDWELL, S COLE F30602-78-C-0078
UNCLASSIFIED RADC-TR-81-250-VOL-2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-81-250, Vol-II (of three)	2. GOVT ACCESSION NO.: AD-A110 272	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): SABERS STAND-ALONE ADIC BINARY EXPLOITATION RESOURCES SYSTEM		5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report April 78 - January 81
		6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Albert J. Franklin, Randy L. Caldwell, Scott Cole, Thomas L. McGibbon, Kathy H. Michel, James R. Wilson		8. CONTRACT OR GRANT NUMBER(s): F30602-78-C-0078
9. PERFORMING ORGANIZATION NAME AND ADDRESS: PAR TECHNOLOGY CORPORATION, 228 Liberty Plaza, Rome NY 13440		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 64750F 19550114
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRDT), Griffiss AFB NY 13441		12. REPORT DATE: September 1981
		13. NUMBER OF PAGES: 302
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Same		15. SECURITY CLASS. (of this report): UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Garry W. Barringer (IRDT)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Data Base Management Systems
Transaction Processing
Graphic Systems
Sperry-Univac 1652 Terminal
ADCOM Applications
Orbital Mechanics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The SABERS development effort has been to design and implement a cohesive system for the Aerospace Defense Command (ADCOM) to provide an upgraded and improved analyst capability for the ADCOM Intelligence Center (ADIC) and its missions. In addition, SABERS has developed system software (such as a data base management system, a user interface, and a graphics package) to support current and future ADIC application needs. The SABERS application system provides an upgraded capability for the ADIC analyst, utilizes

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

data currently available within the ADIC, is general and flexible in its use, and is designed to minimize the amount of information the analyst has to enter into the system. The application functions implemented are built around a set of ten (10) data bases which are directly accessible by the analysts. The functions include a number of numerical and graphical applications. System software that is part of the current SABERS implementation includes a data base management system (DBMS), a user interface, and a graphics package. Goals reached in the DBMS development include the ideal that the application programmer's software interface to the SABERS DBMS should be at a high enough level such that the programmer can easily describe to the DB S the information content of his data base, easily create the data base and then easily access the information in the data base. Furthermore, powerful data base search and retrieval capabilities are part of the DBMS. Data base management applications provide a generalized capability for examining, updating, adding to or deleting information contained in the data bases. Goals realized by the user interface subsystem include the ideal that the application programmer's software interface to the SABERS user's terminal is to be at a high enough level that the programmer does not have to concern himself with the idiosyncrasies of the terminal. It should be easy for the programmer to describe to this interface the format of the display to be presented to the user. It should be easy for the interface to present the display to the user and to receive inputs from him. Finally, it should be easy for the programmer to access the inputs. The primary goal of the graphics package which is realized in SABERS is the ideal that an application programmer should be able to describe a picture to the graphics package using data values he understands. The graphics package performs all the necessary transformations to map a picture from the user's coordinate system into the terminal's coordinate system. The graphics package is also as terminal-independent as possible. A major part of the SABERS effort was the development of software for the Sperry-Univac 1652 terminal. This development involved designing and implementing code within the 1652 to interface it with the SABERS computer system (the VAX 11/780) as well as designing and implementing the code to control the terminal.

APPENDIX A

SPACE AND MISSILE ANALYST USER MANUAL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION TO SABERS

A.1 INTRODUCTION TO SABERS

The analyst is intended to be an integral part of the SABERS system. This means that the analyst is responsible for directing the sequence of operations and for controlling the inputs to the applications. The analyst is aided by SABERS in controlling application input, since parameters are maintained by the system which assist the applications in supplying default information. The extensive use of default information, in conjunction with the screen editing tools provided by SABERS, reduces the workload of the analyst without reducing his control over the system.

A.1.1 Complementary Structure

SABERS consists of software routines in six functional areas: applications, map drawing, user interaction, graphics, data base management, and terminal control. The software developed in each of these areas is designed to interact with and support all of the other areas, and to provide capabilities which may be easily modified and expanded. This combination of mutually interactive and removable functional areas is called complementary structure.

SABERS was developed to provide a set of tools for demonstrating improvements to space and missile analysts at the ADCOM Intelligence Center (ADIC). The complementary structure provides a testbed for evaluating the effectiveness of computer aided research and analysis. In providing this service, the complementary structure functions as a preliminary system design; thus, for simplicity, it is referred to as the SABERS system throughout this manual.

A.1.2 Transaction Processing

In SABERS, the applications require extensive interaction with the analyst. Rather than present the analyst with a series of questions attempting to direct the input of the data required by the application, SABERS uses the concept of transaction processing. In transaction processing, the analyst is presented with a form on the monitor which displays all the information the task requires. This form is called a screen. Space is provided near the information requested for the analyst to type his inputs. These spaces are referred to as fields. Depending on the application, a field may contain default data when the screen is presented to the analyst.

Besides the capability of seeing all the inputs required at one time, another advantage of transaction processing is the flexibility of input. Defaults may be changed by typing over the information presented in the field. Blank fields are recognized as fields for which no values have been entered, and for some applications, lists and ranges may be allowed as responses. Finally, the analyst's choice of inputs remains on the monitor after the data has been sent and is being used by the application to generate its output.

SABERS provides many editing tools to facilitate preparing the screen for sending the inputs required back to the application. These tools are discussed in Section A.2.3.

The types of inputs allowed in a field are application dependent. For example, lists and ranges are only valid in building an assertion for a data base review function. A single value in an input field is always acceptable. Blank fields have different meanings as input depending on the application and the intended use of the input value. The uses and meanings may be summarized as:

Application use	Meaning of Blank Field
Data base search assertion	Match all values
Modify data base input	Value unknown; blank field inserted in data base
Numerical analysis input	Decimal value set to zero Name field unused

When the analyst is satisfied with the contents of the screen (either as presented by the application or after editing), he sends the data to the application by pressing the SEND control key. If an error is detected in the data (for example, a wrong data type or a number out of range), the monitor is cleared and a descriptive error message is written on the monitor. About 3 seconds later, the screen is redisplayed with blinking question marks filling the field that is in error. The same mechanism is followed if the application detects that a blank field must contain some input. The analyst response should be either to move to the designated field and enter the correct data or to exit the editing session by pressing the ABORT soft key. A list of possible error messages, with the suggested analyst response, is presented in Section A.8.

Once the data input to the application does not result in entry errors, then the application will begin processing the data. The result of processing may be to extract new information from the data base and redisplay the screen with new defaults, or to produce output. If the screen is redisplayed with new defaults, the analyst has the same initial options of sending the screen as is, editing the screen before sending it, or aborting the editing session. It should be noted that the application may be aborted at any time during an editing session by pressing the ABORT soft key.

A.1.3 Conventions

In Sections A.3 to A.6, examples of the transaction screens for input and the information screens for output are presented for the current SABERS implementation. In presenting these examples, the following conventions are followed:

1. Textual descriptive information is shown as it appears on the screen.

2. Information, which is output by the application and which may vary across different runs of the same application, is represented by a series of X's.

3. If an application expects input from the analyst for a particular field, this is represented by a series of underscores ("_").

For example, if the display description of a particular option is as follows:

OBJECT BEING OBSERVED ____________

SENSOR ID XXXXXX

LAUNCH ID XXXXXX

"OBJECT BEING OBSERVED", "SENSOR ID", AND "LAUNCH ID" are merely textual information describing the output field to follow and/or the input field that is expected. The underscores after "OBJECT BEING OBSERVED' indicate that input is expected for this field. The X's after "SENSOR ID" indicate that the application outputs the value to this field.

The underscored X's after "LAUNCH ID" indicate that the application outputs default information to this field and that input is expected for this field. If information is not edited in an input field, the value already in that field is transmitted to the application when the SEND control key is pressed. If there is no value in the field when the SEND control key is

pressed, the application is made aware of this fact.

The units of the numerical input and output data are generally degrees for measurements of angles and positions expressed in latitude and longitude, and kilometers for linear distances. If different units are required, the units are prominently displayed on the screen in the textual descriptive information.

A.1.4 SABERS Application Outputs

The types of output possible in SABERS consist of information screens, modified data bases, graphics displays, listing files and line printer listings. Generally, information screens are the output of data base review functions. These functions are discussed in Section A.3.2.

Modified data bases are usually the result of the remaining data base applications (update, add, and delete, discussed in Sections A.3.3, A.3.4 and A.3.5). The modify data base applications may require further information from the analyst. If this is true, the analyst is presented with a screen which may be edited in the same manner as the initial screen was. The supplemental data on the screen must be sent to the application by pressing the SEND control key, or the option may be aborted by pressing the ABORT soft key.

In general, graphic displays are generated by map applications, described in Section A.4, and by the numerical applications, discussed in Section A.6. There are two modes of graphic output by the SABERS applications, new frame and overlay. An application producing graphic output in the new frame mode erases the current graphic display before displaying the output. An application producing graphic output in the overlay mode does not erase the current graphic display first. The output of the overlay mode application is added to the current display. The monitor upon which the graphic output is displayed is selected by the GRAPHICS rocker switch. (See Section A.2.1.)

There are some applications which create a listing file, which may be reviewed at the terminal or printed on the line printer. The listing file may be reviewed at the terminal by pressing the VIEW AT TERMINAL soft key, and responding to the prompt with the name of the file desired. The listing file may be printed on the line printer by pressing the HARDCOPY TO LINE PRINTER soft key, and responding to the prompt with the name of the file desired. The numerical applications which create listing files are presented in Section

A.5. These applications output the name of the listing file created before finishing. The summary review application, described in Section A.3.2.7, automatically prints the listing file it creates on the line printer.

There are two methods for the analyst to request that screen images be automatically printed on the line printer. The first method is invoked by pressing the HARDCOPY THIS SCREEN IMAGE soft key at any time during any screen editing session. This causes a copy of the edited screen to be printed when the SEND control key is pressed. The second method is invoked as an utility application at any time the analyst is not in a screen editing session by pressing the PRINT LAST SCREEN IMAGE soft key. This causes the last screen image displayed to be printed on the line printer.

A.2 USING THE S-U 1652

The terminal which has been selected for Space and Missile analysts to use while communicating with SABERS is the Sperry-Univac 1652 Dual Monitor Terminal. It is referred to as the S-U 1652 throughout this manual. Before an analyst can use SABERS intelligently, he must be familiar with the S-U 1652.

This section presents the basic information necessary for using the S-U 1652; therefore it should be read thoroughly before proceeding any further with this manual. The remainder of this chapter is divided into four sections:

A.2.1 Presents the layout of the terminal and provides an introduction to the various keys and screens.

A.2.2 Gives the instructions for logging on and logging off the terminal.

A.2.3 Tells how to do the screen editing required for transaction screen processing. The functions of the control keys and the programmable soft keys used in screen editing <u>only</u> are described here.

A.2.4 Describes the functions of the remaining control keys.

NOTE

SABERS currently uses 52 of the 60 programmable soft keys available on the S-U 1652 terminal. Only seven of these keys are used in screen editing; the others are used for various functions in the data base, map, listing-generating analysis, and graphic analysis applications. <u>Only</u> those keys used in screen editing are described in this chapter. The others are described in the appropriate Sections A.3 through A.6. In addition, Section A.7 provides an alphabetical reference guide to all control keys and programmable soft keys, with a brief description of their functions.

A.2.1 S-U 1652 Terminal Layout

Figure A-1 is a picture of the S-U 1652 terminal layout. SABERS applications make use of the dual screen monitors, the alphanumeric keyboard, the control key clusters, and the programmable soft keys. In addition, the analyst may use the GRAPHICS rocker switch to control which screen graphic application outputs will appear on.

A.2.1.1 Monitor Screens

By default, the left monitor screen is reserved for transaction processing, leaving the right monitor screen free for graphics applications. In some instances, when the analyst does not require graphics output, both screens can display transaction processing. Regardless of their relative positions, the graphics screen and transaction processing screen can be erased and renewed independently of each other.

A.2.1.2 Rocker Switch

The GRAPHICS rocker switch is the middle one of the three rocker switches at the top of the keyboard. Its normal position is "R", indicating that graphics displays will appear on the right monitor screen. It may be changed to "L" to display graphics on the left screen, or to "OFF" to keep graphics from being displayed. Whether or not graphics information is displayed, it is not changed or lost when the switch is changed.

The other two rocker switches, INTERNAL VIDEO and EXTERNAL VIDEO, are not used by SABERS. They should be left in the OFF position. Changing their settings will cause the terminal to behave unpredictably. Table A-1 summarizes the permissable rocker switch settings.

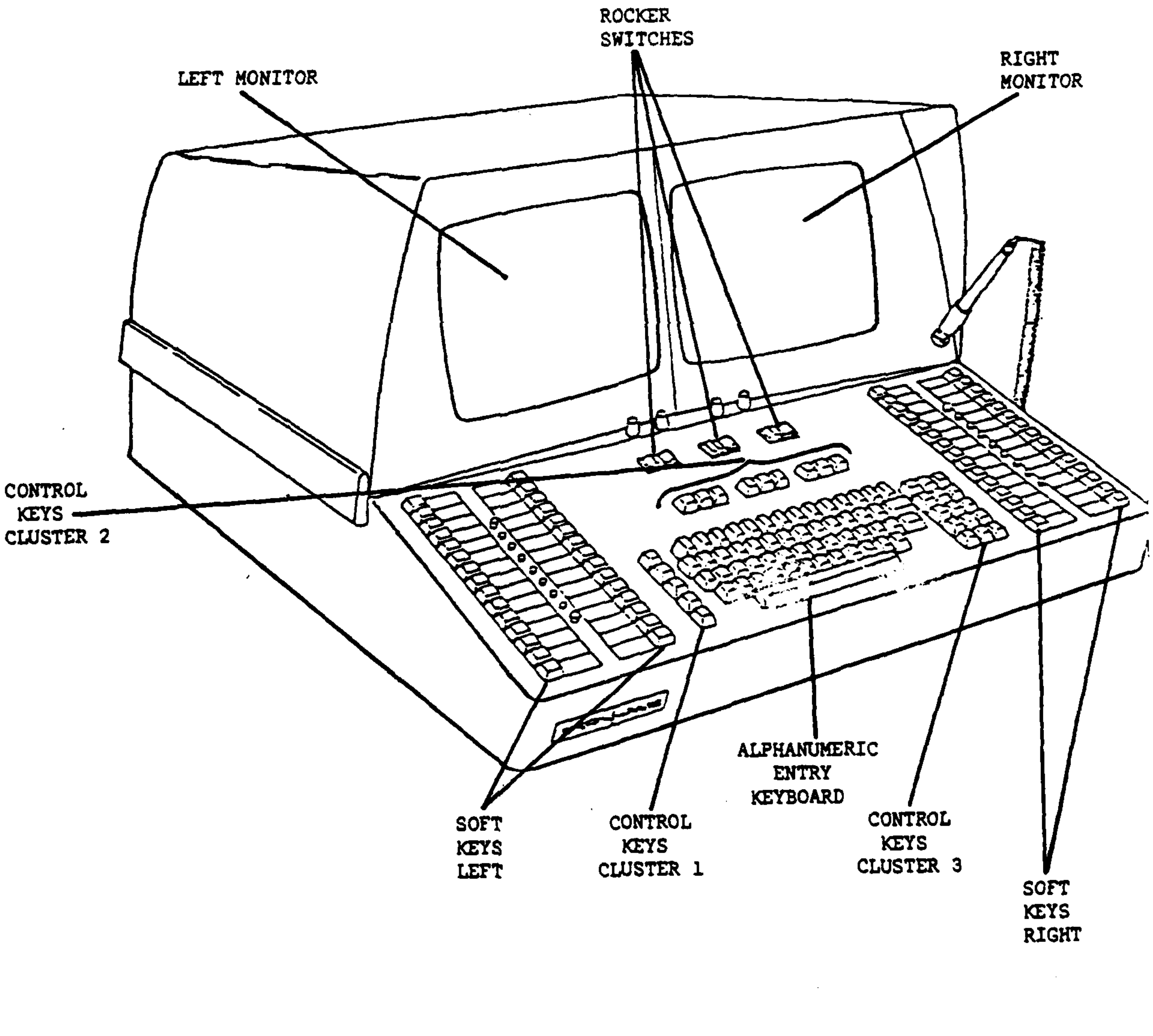


Figure A-1 SPERRY-UNIVAC 1652 Dual Monitor Terminal

Table A-1 Possible Rocker Switch Settings

EXTERNAL VIDEO	
OFF	do not receive signals from external source
GRAPHICS	
L	display graphics on left monitor
OFF	do not display graphics (data is not lost)
R	display graphics on right monitor
INTERNAL VIDEO	
OFF	do not send signals to external receivers

A.2.1.3 Alphanumeric Keyboard

The alphanumeric entry keyboard is a standard typewriter keyboard, and is used in the normal way for entering commands and data. This keyboard is discussed in greater detail in Section A.2.3, "Transaction Screen Editing."

A.2.1.4 Control Keys

The control key clusters are those keys surrounding the keyboard. Their functions are pre-determined; they cannot be changed or reprogrammed. The control keys used in transaction screen editing are discussed in the following section. The remainder of the control keys are described in Section A.2.4, as well as in Chapter A.7 "Reference Guide to Control and Programmable Soft Keys."

A.2.1.5 Programmable Soft Keys

The soft keys are predefined by the SABERS software. Each soft key is used to transmit a stored string of characters, called a program, to the computer when the key is pressed. The computer receives the program just as though the user had typed it in from the keyboard. Currently SABERS uses 52 of the 60 available soft keys; those keys which are defined are the ones for which the lightable menu slot is lit.

Figures A-2 and A-3 show the currently defined soft keys. The numbers in parentheses shown in the margin beside each key indicate the section of this manual where the key is discussed. For example, the SUMMARY key is described in Section A.3.2.7. All the keys used in transaction screen editing are described in Section A.2.3; the others are described in Sections A.3 through A.6, depending on the type of application they are used in. In addition, Section A.7 provides an alphabetical listing of the programmable soft keys with their functions.

WARNING

The definitions of the programmable soft keys should not be changed by the user. In particular, the analyst should be very careful not to press the control key labeled LOCAL CLEAR SOFT KEYS. If this key is pressed by accident it will erase the definitions of all the soft keys, and they will have to be restored by the system maintainer before the terminal will be useful again.

(A.3.2.7)	SUMMARY	CURRENT LAUNCH REVIEW	(A.3.2.6)
(Undefined)			(Undefined)
(Undefined)		LAUNCH FOLDERS	(A.3.2.2)
(A.3.4)	ADD A NEW RECORD	LAUNCH VEHICLES	(A.3.2.2)
(A.3.3.1)	UPDATE AN EXISTING RECORD	LAUNCH SITES	(A.3.2.2)
(A.3.5)	DELETE AN EXISTING RECORD	BLUE RADAR SYSTEMS	(A.3.2.2)
(Undefined)		BLUE SPACEBORNE SYSTEMS	(A.3.2.2)
(A.1.4)	PRINT LAST SCREEN IMAGE	SOVIET SOB	(A.3.2.2)
(A.3.2.3)	EXAMINE NEXT RECORD	RED SUPPORT FACILITIES	(A.3.2.2)
(A.3.2.4)	EXAMINE CURRENT RECORD	RADAR INPUTS	(A.3.2.2)
(A.3.2.5)	EXAMINE PREVIOUS RECORD	IR SENSOR INPUTS	(A.3.2.2)
(A.2.3)	RETYPE THE SCREEN	POLYNOMIAL INPUTS	(A.3.2.2)
(A.2.3)	HARDCOPY THIS SCREEN IMAGE	ABORT	(A.2.3)
(A.2.3)	TOP OF PAGE	BOTTOM OF PAGE	(A.2.3)
(A.2.3)	ERASE THIS FIELD	INSTRUCTIONS	(A.2.3)

FIGURE A-2
LEFT SOFT KEYS

(A.3.3.2)	SELECT LAUNCH ID		(Undefined)
(A.3.3.2)	SELECT PAYLOAD ID	ALAPP PLOT	(A.6.3)
(Undefined)		SPIDER PLOT	(A.6.2)
(A.4.1.1)	DISPLAY A WORLD MAP	AUTOMATICALLY CYCLE	(A.6.4)
(A.4.2.1)	OVERLAY CURRENT LAUNCH POINT	TWO SENSOR ANALYSIS	(A.6.5)
(A.4.2.2)	OVERLAY LAUNCH SITES		(Undefined)
(A.4.2.3)	OVERLAY RED SUPPORT FACILITIES	GENERATE THREAT WINDOWS	(A.5.1)
(A.4.2.4)	OVERLAY BLUE RADAR COVERAGE		(Undefined)
(A.4.2.7)	OVERLAY RADARS VS. ORBIT	LIST RADARS VS. ORBIT	(A.5.2)
(A.4.2.8)	OVERLAY SATELLITE RECONNAISSANCE	LIST SATELLITE RECONNAISSANCE	(A.5.3)
(A.4.2.5)	OVERLAY GROUND TRACE	RADARS VS. ORBIT	(A.5.4.1)
(A.4.2.6)	OVERLAY TIME MARKS ON GROUND TRACE	SATELLITE RECONNAISSANCE	(A.5.4.2)
(A.4.1.2)	DRAW POLITICAL BOUNDARIES	DRAW MAP GRID	(A.4.1.3)
(A.6.1)	ZOOM ON LAUNCH SITE	HARDCOPY TO LINE PRINTER	(A.5.5.1)
(A.4.1.4)	REDRAW MAP ONLY	VIEW AT TERMINAL	(A.5.5.2)

FIGURE A-3
RIGHT SOFT KEYS

A.2.2 Logging On And Off

To begin a SABERS session, you must make contact with the computer, a process known as "logging on." When the SABRES session is over, you must signal this fact to the computer, or "log off." Both procedures are simple.

A.2.2.1 Log On

To log on, sit down at the terminal and press the RETURN key. The computer will respond with the prompt:

USERNAME:__

Type in the name which your system manager has assigned to you. Press the RETURN key. The response will be:

PASSWORD:__

Type in your assigned password. It will not appear on the screen. The computer will respond with the message:

PLEASE ENTER YOUR INITIALS

After you have entered your initials and hit RETURN, the computer will check to see whether you are an authorized SABERS user. When it has complete verification, the lights will come on beside the programmable soft keys, and SABERS will print several system messages on the screen, including:

PREVIOUS LOGICAL NAME ASSIGNMENT REPLACED

% RUN-S-PROC_ID, identification of created process is XXXXXXXX

From this point, you can invoke any SABERS application, do transaction screen processing, update data bases, or perform any other SABERS function. At the end of the session, you will log off.

A.2.2.2 Log Off

To end a SABERS session, first complete any transaction screen editing or applications you may be running. Enter a PURGE command to remove multiple copies of files from your area. This is a necessary precaution, since many of the applications create files during their operation. Purging your files at the end of each SABERS session will keep your memory area from filling up. Once the purge is executed, enter the command:

LOG

The lights beside the programmable soft keys will go out, and the computer will respond with a message indicating that it has logged you off, and showing the time and date. For example:

```
                    HAVE A GOOD DAY
TLF       logged out at 2-APR-1981   11:35:52. 91
```

A.2.2.3 Unexpected Response

Occasionally you may receive an unexpected response to your attempt to log on or off. For example, entering your password may produce the response:

LOGIN - user validation error

This indicates that you have entered your password incorrectly. Press RETURN to get the USERNAME response and try again.

Generally, the best way to deal with an unexpected response is to try again, making sure that you are doing each step correctly. If, after two or three tries, you still do not get the right response, you should consult the system maintainer.

A.2.3 Screen Transaction Editing

The screen editing features of the terminal are designed to provide maximum utility and clarity of results. In this section, the effect of the terminal keys, the control keys, and the soft keys relevant to screen editing is presented.

The concept of a screen data field has been introduced in Section A.1.2. The position of the cursor (the blinking rectangle maintained on the monitor) indicates to the analyst the field currently available for editing, and the character position in that field. Each field contains room for a certain number of characters. This number of characters is referred to as the width of the field.

The analyst may determine whether the screen present on the monitor may be edited or not by the location of the cursor on the monitor. If the cursor is in some data input field in the screen, editing may occur. If the cursor is located outside the screen at the bottom of the monitor (normally as a result of pressing the SEND control key), the screen is not available for editing. The time between the beginning and the end of the ability to edit the screen is referred to as an editing session. An editing session may be terminated by pressing the SEND control key or the ABORT soft key.

WARNING

> The only time that an application may be aborted is during the screen editing session, by pressing the ABORT soft key. The analyst is warned to never abort an application by pressing the EXIT control key. If the EXIT control key is pressed to stop execution of the application at any time, the system environment will probably be destroyed. The environment will then have to be restored by the system maintainer before any further SABERS work can be done.

During any screen editing session, the message "END OF SCREEN - HIT HOME OR SEND" is appended to the end of the screen on the monitor when some of the editing keys are pressed while the cursor is in the last screen field. When the message is printed, the cursor is left at the end of the message. The analyst may respond to this message in one of three ways. The first is by pressing any character key, the carriage return key, or any of the control keys appropriate for editing (other than SEND). This causes the cursor to be positioned at the beginning of the first field of the screen. The second acceptable response is pressing the SEND control key. This causes the screen data to be transferred to the application. The third response is pressing the ABORT soft key. Pressing the ABORT soft key at any time during an editing session will cause the screen editing session and the application to be aborted.

If, during a screen editing session, an error is detected or the analyst requests information, the monitor is cleared, a message may be displayed, and then the screen is redisplayed. The cursor is repositioned at the data field it was in when the circumstance occurred. If an error is detected after the SEND control key is pressed, the monitor is cleared, an error message is displayed for about 3 seconds, the screen is redisplayed, and the cursor is positioned at the beginning of the first field of the screen. In this case, the analyst must move the cursor to the field in error, which is designated by the blinking question marks. A reference guide to SABERS error messages and the correct response to each is provided in Section A.8.

Note that a blank field may be interpreted as a "don't care" response. This is possible because the application is informed that no values have been input for each field which is blank when the SEND control key is pressed.

CAUTION

> The analyst is cautioned against pressing non-editing soft keys during a screen editing session. The effect of pressing one of these soft keys during editing is to fill

one or more screen data fields with the characters which make up the program stored for that soft key. If this happens, the analyst must either retype the lost inputs, or abort the editing session by pressing the ABORT soft key, and restart the application.

A.2.3.1 Alphanumeric Entry Keyboard Keys and RETURN

The keyboard character keys are discussed separately from the carriage return key RETURN. This separation is due to the alphanumeric data entering nature of the character keys, as opposed to the control nature of the RETURN key (identical to that of the NEXT FIELD and "RIGHT-ARROW" control keys described in the next subsection.)

Character Keys

The keyboard keys which are acceptable for use by the analyst during screen editing are the alphabetic characters (upper and lower case), the digits (0 to 9), and the special characters, including plus sign ("+" for positive numbers), minus sign ("-" for negative numbers), decimal point ("." for real numbers), comma ("," to separate items in a list or range specification), left parenthesis ("(" to designate the lowest value in a range), and right parenthesis (")" to denote the largest value in a range). If an illegal character is entered during the editing session, the monitor is cleared, the error message "SORRY. THAT WAS AN ILLEGAL CHARACTER." is displayed on the monitor for about 3 seconds before redisplaying the screen, and the illegal character is deleted.

Each field on the screen can hold only a certain number of characters. When a field is completely filled by the characters typed in by the analyst, the cursor is automatically positioned to the beginning of the next field. If the last field is completely filled, the message "END OF SCREEN - HIT HOME OR SEND" is appended to the screen text. Any character response will cause the cursor to go to the beginning of the first field.

RETURN

The RETURN key is pressed to indicate the end of an item when the character string being typed into a field is smaller than the field width. When RETURN is pressed to terminate the item, the rest of the field is erased, and the cursor is positioned at the beginning of the next field.

The RETURN key is also pressed to move the cursor to the beginning of the next field. When RETURN is pressed without having entered a character in the field, the content of the field is not altered, and the cursor is moved to the beginning of the next field.

If the RETURN key is pressed in the last field, the message "END OF SCREEN - HIT HOME OR SEND" is appended to the screen text. Pressing RETURN again will move the cursor to the beginning of the first field of the screen.

A.2.3.2 Control Keys

The control keys have been introduced in Section A.2.1. In this section, each of the control keys used in transaction processing is discussed in more detail. Any of the control keys not mentioned in this section should not be used during screen editing, either because they hinder the editing process at best or may destroy the system environment at worst. The control keys used for other functions than editing are described in subsection A.2.4.

NEXT FIELD

Pressing the NEXT FIELD control key moves the cursor to the beginning of the next field. If NEXT FIELD is pressed without having entered a character in the field, the content of the field is not altered before the cursor is moved. If NEXT FIELD is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the cursor is moved.

If the NEXT FIELD key is pressed in the last field, the message "END OF SCREEN - HIT HOME OR SEND" is appended to the screen text. Pressing NEXT FIELD again will move the cursor to the beginning of the first field of the screen.

SEND

Pressing the SEND control key initiates the transfer of the data on the screen to the application. The cursor indicates that no further editing of the screen may occur by moving to the free space outside the screen at the bottom of the monitor. The contents of the screen are printed on the line printer if the HARDCOPY THIS SCREEN IMAGE soft key was pressed at any time during the editing session. The screen data is checked for errors. If an error is found, the monitor is cleared, an appropriate error message is printed, and the screen is redisplayed with blinking question marks filling the field in which the error occurred. If no error is detected by the screen editor, the data is transferred to the application.

The application understands that no values are input for each field that consists of all blanks. If SEND is pressed without having entered a character in the field, the content of the field is not altered before the data transfer is initiated. If SEND is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the data transfer is initiated.

RUBOUT

Pressing the RUBOUT control key deletes the last character typed into the field. If no characters have been typed, pressing RUBOUT has no effect. If the RUBOUT key has been used to delete all the characters typed into the field, and the original item in the field is required, the analyst should press the RETYPE THE SCREEN soft key to verify the content of the field. RUBOUT and CHAR DEL are functionally identical.

CHAR DEL

Pressing the CHAR DEL control key deletes the last character typed into the field. If no characters have been typed, pressing CHAR DEL has no effect. If the CHAR DEL key has been used to delete all the characters typed into the field, *and the original item in the field is required*, the analyst should press the RETYPE THE SCREEN soft key to verify the content of the field.

UP-ARROW

Pressing the "UP-ARROW" control key moves the cursor to the beginning of the first field on the previous line. If "UP-ARROW" is pressed without having entered a character in the field, the content of the field is not altered before the cursor is moved. If "UP-ARROW" is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the cursor is moved.

If the "UP-ARROW" key is pressed in the first line, there is no effect. If the message "END OF SCREEN - HIT HOME OR SEND" has just been appended to the screen, pressing "UP-ARROW" will move the cursor to the beginning of the first field of the screen.

LEFT-ARROW

Pressing the "LEFT-ARROW" control key moves the cursor to the beginning of the first field on the left, retreating to the previous line if necessary. If "LEFT-ARROW" is pressed without having entered a character in the field, the content of the field is not altered before the cursor is moved. If "LEFT-ARROW" is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the cursor is moved.

If the "LEFT-ARROW" key is pressed in the first field, there is no effect. If the message "END OF SCREEN - HIT HOME OR SEND" has just been appended to the screen, pressing "LEFT-ARROW" will move the cursor to the beginning of the first field of the screen.

HOME

Pressing the HOME control key moves the cursor to the beginning of the first field in the first line of the screen. If HOME is pressed without having entered a character in the field, the content of the field is not altered before the cursor is moved. If HOME is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the cursor is moved. If the message "END OF SCREEN - HIT HOME OR SEND" has just been appended to the screen, pressing "HOME" will move the cursor to the beginning of the first field of the screen.

RIGHT-ARROW

Pressing the "RIGHT-ARROW" control key moves the cursor to the beginning of the next field on the right, advancing to the next line if necessary. If "RIGHT-ARROW" is pressed without having entered a character in the field, the content of the field is not altered before the cursor is moved. If "RIGHT-ARROW" is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the cursor is moved.

If the "RIGHT-ARROW" key is pressed in the last field, the message "END OF SCREEN - HIT HOME OR SEND" is appended to the screen text. Pressing "RIGHT-ARROW" again will move the cursor to the beginning of the first field of the screen.

DOWN-ARROW

Pressing the "DOWN-ARROW" control key moves the cursor to the beginning of the first field of the next line. If "DOWN-ARROW" is pressed without having entered a character in the field, the content of the field is not altered before the cursor is moved. If "DOWN-ARROW" is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the cursor is moved.

If the "DOWN-ARROW" key is pressed in the last line, the message "END OF SCREEN - HIT HOME OR SEND" is appended to the screen text. Pressing "DOWN-ARROW" again will position the cursor to the beginning of the first field of the screen.

A.2.3.3 Predefined Soft Keys

The soft keys mentioned in this section are designed to be provide extra capabilities beyond those supplied by the control keys. These soft key programs are predefined by SABERS, and should not be altered by the analyst. In particular, the analyst should never press the LOCAL CLEAR SOFT KEYS control key.

INSTRUCTIONS

Pressing the INSTRUCTIONS soft key results in clearing the monitor and displaying the screen represented in Figure A-4. This screen attempts to document the editing features for reference. Pressing RETURN causes the monitor to be erased and the original data screen to be redisplayed.

If INSTRUCTIONS is pressed without having entered a character in the field, the content of the field is not altered before the monitor is cleared. If INSTRUCTIONS is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the monitor is cleared.

USE THE FOLLOWING SYMBOLS TO PERFORM THE SPECIFIED ACTIONS:

CARRIAGE RETURN = GO TO NEXT FIELD.
DOWN-ARROW = GO DOWN ONE LINE.
UP-ARROW = GO UP ONE LINE.
LEFT-ARROW = BACK TO PREVIOUS INPUT FIELD.
RIGHT-ARROW = GO TO NEXT FIELD.
HOME = GO TO TOP LINE.
BOTTOM = GO TO BOTTOM LINE.
ERASE FIELD = ERASE CURRENT FIELD.
INSTRUCTIONS = DISPLAY THESE INSTRUCTIONS.
PRINT SCREEN = PRINT THE CONTENTS OF THIS SCREEN.
ABORT = ABORT EDITING SESSION.
RE-TYPE = RE-TYPE THE SCREEN INCLUDING CORRECTIONS.
SEND = EDITING COMPLETED.

PLEASE NOTE. WHEN YOU HAVE FILLED A
FIELD, THE CURSOR WILL MOVE
TO THE SUCCEEDING FIELD.

HIT CARRIAGE RETURN TO BEGIN EDITING AGAIN.

FIGURE A-4
INSTRUCTION SCREEN

BOTTOM OF PAGE

Pressing the BOTTOM OF PAGE soft key positions the cursor to the beginning of the first field in the last line of the screen. If BOTTOM OF PAGE is pressed without having entered a character in the field, the content of the field is not altered before the cursor is moved. If BOTTOM OF PAGE is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the cursor is moved.

ABORT

Pressing the ABORT soft key results in the termination of both the editing session and the application. The monitor is cleared and the message "EDITING SESSION ABORTED" is displayed. The EXIT control key should not be used to stop a SABERS application at any time because of the danger of system corruption.

ERASE THIS FIELD

Pressing the ERASE THIS FIELD soft key results in replacing the content of the field with all blanks and moving the cursor to the beginning of the next field. Whenever the SEND control key is pressed, the application is informed that no values have been input for each field which contains only blanks.

TOP OF PAGE

Pressing the TOP OF PAGE soft key positions the cursor to the beginning of the first field in the first line of the screen. If TOP OF PAGE is pressed without having entered a character in the field, the content of the field is not altered before the cursor is moved. If TOP OF PAGE is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the cursor is moved.

HARDCOPY THIS SCREEN IMAGE

Pressing the HARDCOPY THIS SCREEN IMAGE soft key at any time during the editing session instructs the system to prepare to automatically print the screen image on the line printer when the SEND control key is pressed. The monitor is cleared, the message "THE COMPLETED SCREEN WILL BE PRINTED AFTER VALIDATION" is displayed for about 3 seconds, and then the screen is redisplayed. Pressing the ABORT soft key negates this option, as the editing session is terminated before the SEND control key can be pressed.

If HARDCOPY THIS SCREEN IMAGE is pressed without having entered a character in the field, the content of the field is not altered before the monitor is cleared. If HARDCOPY THIS SCREEN IMAGE is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the monitor is cleared.

RETYPE THE SCREEN

Pressing the RETYPE THE SCREEN soft key results in the clearing of the monitor and redisplaying the screen. This makes it possible to redisplay a screen inadvertantly erased by the CLR or NEW SCREEN control keys (See Section A.2.4).

If RETYPE THE SCREEN is pressed without having entered a character in the field, the content of the field is not altered before the monitor is cleared. If RETYPE THE SCREEN is pressed after one or more characters have been entered in the field, the remainder of the field is erased before the monitor is cleared.

A.2.3.4 Other Soft Keys

The remainder of the soft keys currently defined by SABERS are used to invoke various applications. They should not be pressed during an editing session. Descriptions of these keys may be found in the appropriate sections, A.3 through A.6, and in the alphabetical reference guide in Section A.7.

A.2.4 Non-Editing Control Keys

This section discusses the remaining control keys, which are not used in transaction screen editing. Figure A-5 shows the locations of all the S-U 1652 control keys. The current meaning of each control key is defined by the internal software of the S-U 1652 terminal.

The actions of control keys are described as either local or global. Local control keys transmit commands which are recognized and acted upon by the terminal; the fact that the key was pressed is not transmitted to the computer. In contrast, global control keys transmit commands to the computer itself, which recognizes and acts upon the commands. Note that all the control keys used in transaction screen editing are global.

Table A-2 summarizes the control keys and their functions. Those marked with an asterix were described in the previous section. Of the remaining keys, three are used by the Space and Missile analyst and the rest are not. These unused keys were designed to aid software developers; because the keys' definitions are part of the terminal software it was not possible to change or delete their operations. Table A-2 lists the operations of these keys, for completeness; however, the use of these keys may hinder or damage the work of SABERS applications. Therefore, the analyst should be careful never to press the following keys:

ALARM	INIT
BOOT	LINE DEL
CNTL	LOCAL CLEAR SOFT KEYS*
DEFINE SOFT KEY	RELEASE DISPLAY
EOF	RESET
ESC	REVIEW LINE
EXIT	SHOW SOFT KEYS
INHIBIT DISPLAY	TRACE

* This key is particularly dangerous to SABERS software.
See the warning under "Programmable Soft Keys" in Section A.2.1.

1. Cntl
2. Reset
3. Define Soft Key
4. Local Clear Graphics
5. Local Clear Soft Keys
6. Trace
7. Show Soft Keys
8. Boot
9. Inhibit Display
10. Release Display
11. EOF
12. Review Line
13. New Screen
14. Exit
15. ESC

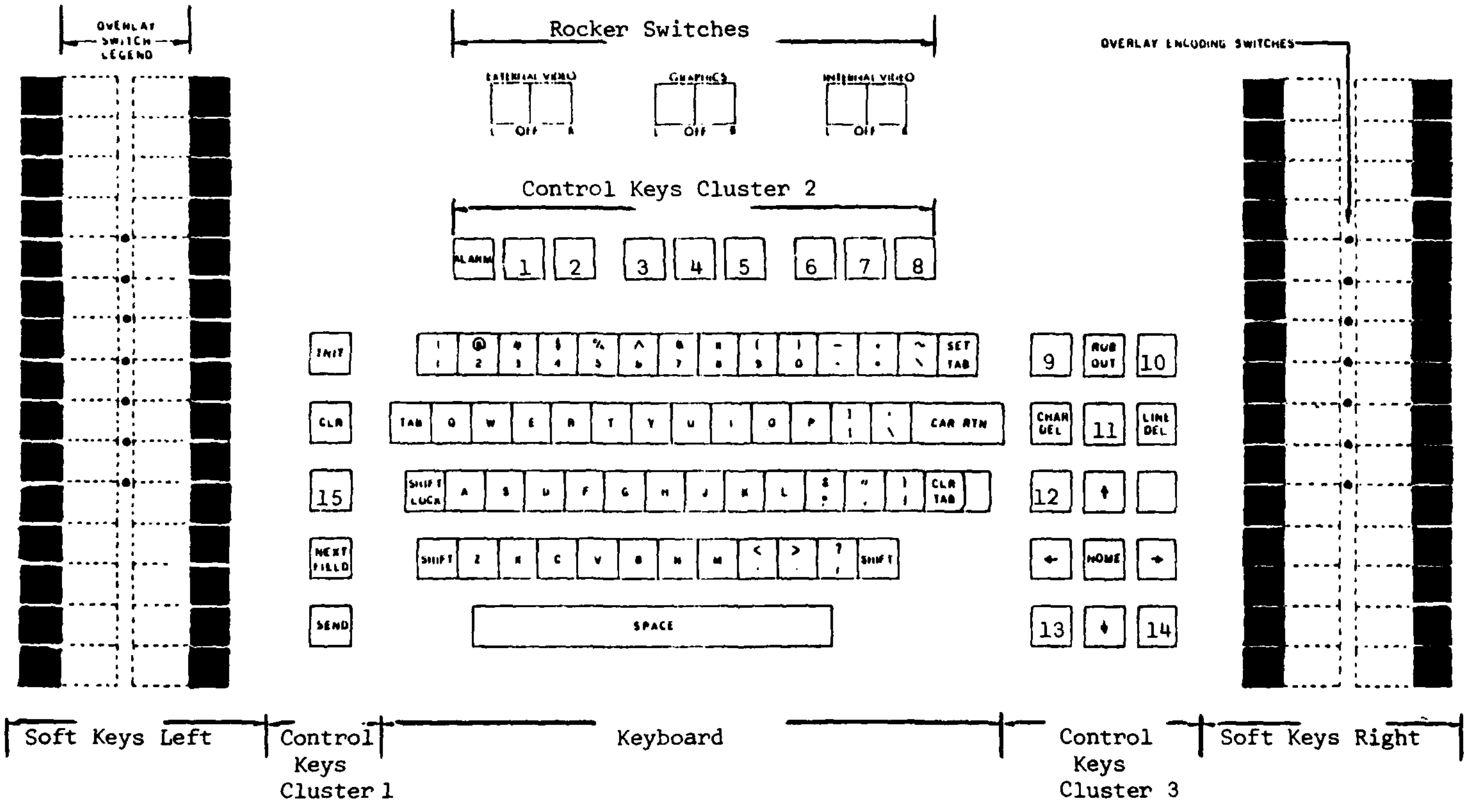


Figure A-5 Key Locations

Table A-2 Control Key Functions

Cluster 1:	to left of keyboard
INIT	must be pressed before RESET or BOOT (local)
CLR	erase text on monitor upon which the cursor appears (local)
ESC	send an escape character to the computer (global)
* NEXT FIELD	advance cursor to next transaction field (global)
* SEND	send the results of transaction to computer (global)
Cluster 2:	above the keyboard
ALARM	not implemented
CNTL	send next keyboard character as a control character (global)
RESET	(after INIT) return terminal to initial booted state (local)
DEFINE SOFT KEY	delineate soft key programming mode (local)
LOCAL CLEAR GRAPHICS	clear the graphics buffer and erase graphics display (local)
LOCAL CLEAR SOFT KEYS	clear all soft key programs (local)
TRACE	show all terminal interaction (local)
SHOW SOFT KEYS	display all soft key programs on other monitor (local)
BOOT	(after INIT) read control program from computer (global)
Cluster 3:	to right of keyboard
INHIBIT DISPLAY	send suspend output symbol (CONTROL-S) to computer (global)
* RUBOUT	delete the last keyboard character typed (global)
RELEASE DISPLAY	send resume output symbol (CONTROL-Q) to computer (global)
* CHAR DEL	delete the last keyboard character typed (global)
EOF	send end of file symbol (CONTROL-Z) to computer (global)
LINE DEL	send delete line symbol (CONTROL-U) to computer (global)
REVIEW LINE	send retype line symbol (CONTROL-R) to computer (global)
* "UP-ARROW"	move cursor up to previous transaction line (global)
* "LEFT-ARROW"	move cursor left to previous transaction field (global)
* HOME	move cursor to first transaction field (global)
* "RIGHT-ARROW"	move cursor right to next transaction field (global)
NEW SCREEN	erase text on, and move cursor to, other screen (local)
* "DOWN-ARROW"	move cursor down to next transaction line (global)
EXIT	send stop execution symbol (CONTROL-Y) to computer (global)

The control keys which are used by the analyst are CLR, LOCAL CLEAR GRAPHICS, and NEW SCREEN. The action of each of these keys is local.

CLR

Pressing this key will erase the text on the monitor where the cursor is currently located. Only non-graphics text is cleared; graphics are not affected.

LOCAL CLEAR GRAPHICS

Pressing this key will clear graphics displays from the monitor. Non-graphics text is not affected.

NEW SCREEN

Pressing this key will clear the text from, and move the cursor to, the other monitor. Only non-graphics text is cleared; graphics are not affected.

If CLR or NEW SCREEN is pressed inadvertantly during an editing session, pressing the RETYPE THE SCREEN soft key will restore the display. If the LOCAL CLEAR GRAPHICS is pressed inadvertantly, the analyst must run the pertinent graphics applications again to restore the display.

A.3 DATA BASE APPLICATIONS

All the information required by the analyst and by the system is maintained in data bases. The analyst-maintained data bases currently available in SABERS are described in Section A.3.1. The system-maintained data bases are briefly mentioned here, since they are not directly accessible by the analyst.

The system-maintained data bases store information used by SABERS to aid the analyst, including the map data (coastlines and political boundaries), the current launch event identification number, and the last record reviewed. The map coordinates stored in the "map data" system data base are used when drawing maps. The current launch identification number is written in the "launch id" system data base when the analyst selects the launch identification number or the payload identification number (see Section A.3.3.2). The launch identification number is used by the SABERS applications to provide default information for the analyst at the beginning of the screen editing session. The last record reviewed (data base and record number) is stored in the "last review" system data base when a review function is executed by the analyst. The last record reviewed is used by the data base maintenance applications to provide default information to the analyst for the screen editing session. The last record reviewed is also used by the Next, Current, and Previous review functions to determine which record should be retrieved from the data base.

The data base maintenance applications provide the analyst with a generalized management capability on the analyst-maintained data bases described in Section A.3.1. These applications include generalized review functions (review data base, Next, Current, Previous, and Summary) described in Section A.3.2, update functions (Update Data Base, Select Launch ID or Payload ID) presented in Section A.3.3, and the Add and Delete Record functions described in Sections A.3.4 and A.3.5.

DATA BASE APPLICATIONS

In the multi-user environment of SABERS, it is possible that a data base record may be deleted between the time that an application retrieves the record and the time that the analyst attempts to display, modify or delete the deleted record. This may occur for any rocord in any application. If the record required by the application has been deleted by another user, the application will detect its absence, clear the monitor, and output the message "RECORD # XX OF YOUR LAST REVIEW HAS BEEN DELETED" before exiting.

A.3.1 Description of Current Data Bases

The analyst-maintained data bases contain data that has been determined to be important to the Space and Missile Intelligence analyst. The data in these data bases may be entered, altered, reviewed and deleted by the analyst using the SABERS data base maintenance applications described in Sections A.3.2 to A.3.5.

A.3.1.1 Event Summary Data Bases

The event summary data bases provide an overview or summary of all the information associated with each launch event. There currently is one event summary data base in SABERS, the "launch folder" data base.

The "launch folder" data base contains a summary of the information associated with a space or missile launch. The contents of one record of this data base is presented in Table A-3. The launch folder acts as the central file linking associated information for each launch event. Included in the launch folder is the unique identification number for each launch event, the launch location, the launch trajectory, the reentry location, the reentry trajectory, and the identification of each space object associated with the event. The launch identification number is assigned by the analyst when he selects an unused launch id number in the Select Launch ID update function in response to a new event.

The launch identification number makes it possible for an application to access the information in the launch folder it requires for presentation as default information in screen editing. The "launch folder" data base should be maintained by the analyst throughout the life of an event to ensure that this information is current and available.

Table A-3 Launch Folder Data Base

Launch Identification Number

Prelaunch Information

Launch Date (Month, Day, Year)

Launch Time (Hour, Minute, Second)

Launch Position (Latitude, Longitude, Altitude)

Launch Site Name

Launch Pad Name

Launch Confirmation Sources

Launch Vehicle

Launch Azimuth

Launch Inclination

Reentry Position (Latitude, Longitude)

Reentry Date (Month, Day, Year)

Reentry Time (Hour, Minute, Second)

Reentry Azimuth

Reentry Inclination

Reentry Confirmation Source

Type of Event (Space or Missile)

Threat or No Threat Classification

Payload Mission

Target Satellite Identification (For ASAT only)

Launched Satellite Identification Number

Other Associated Objects Identification Number (e.g. tank)

Mission Remarks

A.3.1.2 Characteristics and Capabilities Data Bases

The analyst may find it desirable to learn about and understand the characteristics or capabilities of the different vehicles and facilities that may be involved when an event occurs. SABERS provides five data bases which enable him to perform this investigation.

Launch Vehicles

The "launch vehicle" data base describes the characteristics and capabilities of all launch vehicles. The contents of one record of this data base are presented in Table A-4. The information for each launch vehicle includes the launch vehicle name, the payload mission, the payload orbital characteristics, and the IR profile data associated with the launch vehicle-payload mission pair.

The analyst creates a new record in the data base every time the characteristics and capabilities of a new launch vehicle-payload mission pair is determined. The analyst updates the record every time a new characteristic or capability for an existing launch vehicle-payload mission pair is determined.

Launch Sites

The "launch site" data base describes the characteristics and capabilities of each launch pad at each launch site. The contents of one record of this data base are presented in Table A-5. The information stored for each launch site pad includes the launch site name, launch pad name, launch pad identification number, launch pad type, and launch pad location. Also included is a list of launch vehicles and missions capable of being launched from the pad.

Table A-4 Launch Vehicle Data Base

Launch Vehicle Name

Payload Mission

Payload Orbital Characteristics

Maximum Payload Weight

Time vs. Intensity Profile

Azimuth vs. Elevation Profile

Remarks

Table A-5 Launch Site Data Base

Launch Site Name

Launch Pad Name

Launch Pad Type (Space or Missile)

Launch Pad BE Number

Launch Site Pad Location (Latitude, Longitude, Altitude)

Launch Vehicle Capabilities

Missions Capable of Being Launched

The analyst creates a new record in the data base every time a new launch pad is made operational. The analyst updates the record if the characteristics or capabilities for an existing launch site pad change.

Tracking and Receiving Support Facilities

The "tracking facilities" data base describes the characteristics and capabilities of the Red space launch support facilities. The contents of one record of this data base are presented in Table A-6. The information stored for each facility includes the facility name, the facility type, the facility identification number, the location of the facility, and the characteristics of the support facility.

The analyst creates a new record in the data base every time a new facility becomes operational. If the characteristics or capabilities of an existing facility change, the analyst updates the record for this facility.

Blue Ground Based Sensor Systems

The "Blue radar" data base describes the characteristics and capabilities of the Blue ground based radar systems capable of viewing an event. The contents of one record of this data base are presented in Table A-7. The information stored for each radar includes the sensor name, sensor type, sensor identification number, sensor position and the field of view of the radar.

The analyst creates a new record in the data base every time a new sensor is made operational. The analyst updates the record if the characteristics or capabilities for an existing sensor change.

Table A-6   **Tracking and Receiving Support Facilities Data Base**

Facility Name

Facility Type

Facility BE Number

Facility Location (Latitude, Longitude, Altitude)

Facility Characteristics

Table A-7 Blue Ground-Based Sensor Systems Data Base

Sensor Name

Sensor Type

Sensor SDC Number

Sensor Location (Latitude, Longitude, Altitude)

Range Field

Azimuth Minimum

Azimuth Maximum

Elevation Minimum

Elevation Maximum

Blue Spaceborne Sensor Systems

The "Blue spaceborne sensor" data base describes the current location of Blue spaceborne sensors. The contents of one record of this data base are presented in Table A-8. The information stored for each sensor includes the sensor name, the sensor identification number and its orbital element set at epoch.

The analyst creates a new record in the data base whenever a new sensor becomes operational. The analyst must update the record if the orbit of an existing sensor changes.

A.3.1.3 Order of Battle Data Bases

The order of battle data bases provide the analyst with the ability to review the status of known enemy equipment. The current data base of interest to the space and missile analyst in SABERS is the Soviet space order of battle data base.

The "Soviet ESV status" data base is the Soviet space order of battle data base, and provides the analyst with the status of each Red earth satellite vehicle (ESV). The contents of one record of this data base is presented in Table A-9. The information includes identification numbers, the payload mission, the identification of the ESV's associated launch, the launch time and location, and the characteristics of the payload. Information about current and previous orbits is included in the "ground based sensor inputs" data base discussed under "Raw Data Input Data Bases," Section A.3.1.4.

The analyst creates a new record in the data base every time a new satellite is put into orbit. The analyst updates the record if the characteristics of an existing satellite change.

Table A-8 Blue Spaceborne Sensor Systems Data Base

Sensor Identification number

Sensor Name

Sensor Epoch (Year, Day Number, Hour, Minute, Second)

Sensor Right Ascension

Sensor Eccentricity

Sensor Inclination

Sensor Argument of Perigee

Sensor Mean Anomaly

Sensor Mean Motion

Sensor First Time Derivative of Mean Motion

Sensor Second Time Derivative of Mean Motion

Table A-9 Soviet ESV Status Data Base

Payload Identification Number

Sputnik Number

Series/Number (e.g. COS111)

SPADAT Number

Payload Mission

Associated Launch Identification Number

Launch Site Name

Launch Pad Name

Launch Date (Month, Day, Year)

Launch Time (Hour, Minute, Second)

Payload Life Expectancy/Decay Date (Month, Day, Year)

Estimated Payload Weight

Remarks

A.3.1.4 Raw Data Input Data Bases

The current SABERS system does not include data and message communications support. However, when such support is provided, it is assumed that there will be one or more modules which will capture and preprocess the information. The output of this preprocessing will be the raw data input data bases. The SABERS applications currently assume the existence of three of these data bases (however created) and process them as necessary.

In general, raw data input data bases are very dynamic files. The records will be updated any time new data is reported by the respective sensor through the data and message communications links.

IR Inputs

The "IR inputs" data base contains the time tagged preprocessed information as reported by an infra-red (IR) sensor. The contents of one record of this data base are presented in Table A-10. The information includes the identification number of the sensor observing the event, the sensor's name, the launch identification number, the launch date and time, the launch location, and the sensor observations. The sensor observations consist of time, intensity and the line-of-sight angles, azimuth and elevation .

Ground Based Sensor Inputs

The "ground based sensor inputs" data base contains the time tagged orbital information about all objects in space. The contents of one record of this data base are presented in Table A-11. The information includes the SDC number of the observing sensor, the identification of the object being observed, the associated launch ID of the observed object, and the orbital element set of the observed object. The sensor observations are the epoch and orbital element set of the object being observed.

Table A-10 IR Inputs Data Base

Sensor Identification Number

Sensor Name

Launch Identification Number

Launch Date (Month, Day, Year)

Launch Time (Hour, Minute, Second)

Launch Location (Latitude, Longitude, Altitude)

Sensor Observations

- Time of Observation (Hour, Minute, Second)
- Intensity
- Azimuth
- Elevation

Table A-11 Ground Based Sensor Inputs Data Base

SDC Number of Observing Sensor

Identification Number of Object Being Observed

Object Type (e.g. Payload, Fragment, etc.)

Associated Launch Identification Number

Sensor Observations (Epoch and Orbital elements)

- Epoch (Year, Day Number, Hour, Minute, Second)
- Right Ascension
- Eccentricity
- Inclination
- Argument of Perigee
- Mean Anomaly
- Mean Motion
- First Time Derivative of Mean Motion
- Second Time Derivative of Mean Motion

Polynomial Inputs

The "polynomial inputs" data base contains the time tagged polynomial coefficients of interpolated sensor data. The contents of one record of this data base is presented in Table A-12. The information includes the identification of the observing sensor, the launch event identification number, the launch date and time, the launch position, and the sensor observations. The sensor observations consists of time, the time interval and the polynomial coefficients. The "polynomial inputs" data base is not available for summarizing by the SUMMARY soft key function.

Table A-12 Polynomial Inputs Data Base

Sensor Identification Number

Sensor Name

Launch Identification Number

Launch Date (Month, Day, Year)

Launch Time (Hour, Minute, Second)

Launch Location (Latitude, Longitude, Altitude)

Sensor Observations

- Time (Hour, Minute, Second)
- Interval Length
- X Coefficients
- Y Coefficients
- Z Coefficients

A.3.2 Data Base Review Functions

A data base review function exists for each data base in Section A.3.1, which allows the analyst to examine the records in a particular data base which match a particular set of search criteria. The data base review function only outputs the first record which matches the criteria. The review functions Next, Current and Previous provide the analyst with the ability to examine all the records retrieved by the review function, one at a time. The Next function allows the analyst to examine the next record retrieved, Current allows the analyst to reexamine the current record, and Previous allows the analyst to examine the record previous to the current record being reviewed. These functions are discussed in more detail in Sections A.3.2.4 through A.3.2.6.

The Summary review function allows the analyst to examine, at one time in one listing, all the records in a data base which match the search criteria. The listing shows the values contained in the fields designated by the analyst for each record retrieved. The listing is automatically printed on the line printer. Only the "polynomial inputs" data base is not available to this function. The Summary review function is discussed in more detail in Section A.3.2.7.

A.3.2.1 Entering The Data Base Review Criteria

Within each SABERS data base a set of key fields are identified on which data base searches may be performed. When a review function is selected, the analyst is presented with a transaction screen on the left monitor presenting the list of keys in the data base, and is asked to enter search criteria based on these keys. The rules for entering search criteria are as follows:

1. For each key field, the analyst defines a set of assertions or search criteria.

2. An assertion states that the data base is to be searched for records whose key field equals a particular value and/or lies within a particular range in that field.

3. On entering assertions for a particular key field, assertions are separated by commas.

4. To enter an equals assertion, the analyst merely enters the value for which the data base is to be searched.

5. To enter a range assertion, the analyst enters a left parenthesis followed by the minimum value followed by a comma followed by the maximum value followed by a right parenthesis (e.g., "(100,200)" states search for all records between 100 and 200, inclusive).

6. Assertions within each key field are OR'd together. Assertions between key fields are AND'ed together.

7. All values of the key field are matched by a blank assertion.

For example, suppose that there is a data base in SABERS which describes the characteristics and capabilities of all U. S. automobiles. Furthermore, suppose the data base is set up such that the analyst may examine this data base by searching on the year, make, and model of the automobile (these are the key fields of the data base). The transaction screen presented to the analyst looks like:

```
YEAR

MAKE

MODEL
```

The phrases "YEAR", "MAKE", and "MODEL" are supplied to indicate to the analyst which are the key fields.

Now suppose the analyst wants to examine the characteristics and capabilities of all 1978 Chevrolet Novas. To perform this operation, he would edit the screen to look like:

```
YEAR    1978
MAKE    Chevrolet
MODEL   Nova
```

Now suppose the analyst wants to examine all Chevrolet Novas between 1950 and 1970. To perform this operation, the edited screen looks like:

```
YEAR    (1950, 1970)
MAKE    Chevrolet
MODEL   Nova
```

If the analyst wants to examine all Chevrolet Novas between 1950 and 1970 as well as for 1978, the edited screen may look like:

```
YEAR    (1950, 1970), 1978
MAKE    Chevrolet
MODEL   Nova
        or
YEAR    1978, (1950, 1970)
MAKE    Chevrolet
MODEL   Nova
```

Suppose the analyst is interested in all Chevrolets between 1950 and 1970 as well as for 1978. The edited screen may look like:

YEAR (1950, 1970), 1978

MAKE Chevrolet

MODEL

Finally, if the analyst wants to examine all U. S. automobiles between 1950 and 1960 as well as between 1965 and 1970 as well as 1978, the edited screen may look like:

YEAR (1950, 1960), (1965, 1970), 1978

MAKE

MODEL

or

YEAR (1965, 1970), 1978, (1950, 1060)

MAKE

MODEL

The result of any one of these operations is either an indication that no records exist in the data base satisfying the search criteria or an information screen displaying the first record retrieved which matches the criteria. If no records exist matching the search criteria, the monitor is erased, and the message "THERE ARE NO RECORDS MATCHING THE CONDITIONS." is written on the monitor. If one or more records have been retrieved, the first record is displayed on the monitor. The form of the output information screen is:

1 / 404

YEAR	1978
MAKE	Chevrolet
MODEL	Nova
STYLE	Two Door
OPTIONS	Radio White Sidewalls
ID NUMBER	111-222-333-444

The current record number and the total number of records retrieved are displayed in the upper right hand corner separated by the slash. All the data in the record is displayed in the rest of the information screen. It is possible for both the left and right monitor to be used to display all the data for records which contain more data than can be displayed on one monitor. If the analyst wants to display the contents of record 2, he must run the Next review function. If he instead wishes to review record 404, he must run the Previous review functions. (Both the Next and Previous review functions "wrap around".) If the analyst wishes to redisplay record 1, however, he must run the Current review function.

A.3.2.2 Data Base Review

The data bases may be reviewed according to search criteria as described in Section A.3.2.1. The review functions are initiated by pressing the appropriate predefined soft key.

Launch Folder Review

Pressing the LAUNCH FOLDERS soft key initiates the "launch folder" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-6. The analyst may search on any combination of:

```
LAUNCH ID: ----------------------------------------------------
LAUNCH DATE: MONTH: -------------------------------------------
             DAY:   -------------------------------------------
             YEAR:  -------------------------------------------
LAUNCH TIME: HOUR:  -------------------------------------------
             MINUTE:-------------------------------------------
             SECOND:-------------------------------------------
LAUNCH POSITION: LAT: -----------------------------------------
                 LON: -----------------------------------------
                 ALT: -----------------------------------------
LAUNCH SITE: --------------------------------------------------
LAUNCH PAD:  --------------------------------------------------
LAUNCH VEHICLE:------------------------------------------------
LAUNCH AZIMUTH:------------------------------------------------
LAUNCH INCLINATION: -------------------------------------------
EVENT TYPE (SPACE,MISSILE): -----------------------------------
THREAT OR NOTHREAT: -------------------------------------------
PAYLOAD MISSION:-----------------------------------------------
TARGET SATELLITE ID(FOR ASAT ONLY): --------------------------
LAUNCHED SATELLITE ID: ---------------------------------------
```

FIGURE A-6
LAUNCH FOLDER REVIEW INPUT SCREEN

Launch Identification Number

Launch Date

Launch Time

Launch Position

Launch Site Name

Launch Pad Name

Launch Vehicle

Launch Azimuth

Launch Inclination

Type of Event

Threat or No Threat Classification

Payload Mission

Target Satellite Identification Number

Launched Satellite Identification Number

After the search criteria are sent to the application, all launch folders are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-7. The remaining records may be observed by initiating the Next, Current or Previous review functions.

Launch Vehicle Review

Pressing the LAUNCH VEHICLES soft key initiates the "launch vehicle" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-8. The analyst may search on any combination of:

```
LAUNCH IDENTIFICATION NUMBER XXXXXXXX                              XXXX / XXXX
PRE-LAUNCH INFORMATION XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LAUNCH DATE: MONTH XX  DAY XX  YEAR XXXX
LAUNCH TIME: HOUR XX  MINUTE XX  SECOND XXXXXXXX
LAUNCH POSITION: LATITUDE: XXXXXXX LONGITUDE: XXXXXXX ALTITUDE:XXXXXXX
LAUNCH SITE: XXXXXXXX      LAUNCH PAD: XXXXXXXX
CONFIRMATION SOURCES:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LAUNCH VEHICLE: XXXXXXXX
LAUNCH AZIMUTH: XXXXXXXX LAUNCH INCLINATION: XXXXXXXX
REENTRY LOCATION: LATITUDE: XXXXXX LONGITUDE: XXXXXX
REENTRY DATE: MONTH XX DAY XX  YEAR XXXX
REENTRY TIME: HOUR XX  MINUTE XX  SECOND XXXXXXXX
REENTRY AZIMUTH: XXXXXXXXREENTRY INCLINATION: XXXXXXXX
REENTRY CONFIRMATION SOURCE: XXXXXXXX
EVENT TYPE (SPACE, MISSILE): XXXXXXXX
THREAT OR NOTHREAT: XXXXXXXX
PAYLOAD MISSION: XXXXXXXX
TARGET SATELLITE ID (FOR ASAT ONLY): XXXXXXXX
LAUNCHED SATELLITE ID: XXXXXXXX
OTHER OBJECTS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
REMARKS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

FIGURE A-7
LAUNCH FOLDER REVIEW OUTPUT SCREEN

```
LAUNCH VEHICLE: --------------------------------------------------------------
PAYLOAD MISSION: -------------------------------------------------------------
```

FIGURE A-8
LAUNCH VEHICLE REVIEW INPUT SCREEN

Launch Vehicle Name

Payload Mission

After the search criteria are sent to the application, all launch vehicles are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-9. The remaining records may be observed by initiating the Next, Current or Previous review functions.

Launch Site Review

Pressing the LAUNCH SITES soft key initiates the "launch site" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-10. The analyst may search on any combination of:

Launch Site Name

Launch Pad Name

Launch Pad Type

Launch Pad BE Number

Launch Vehicle Capabilities

Missions Capable of Being Launched

After the search criteria are sent to the application, all launch sites are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-11. The remaining records may be observed by initiating the Next, Current or Previous review functions.

LAUNCH VEHICLE: XXXXXXXX   XXXX / XXXX

PAYLOAD MISSION: XXXXXXXX

ORBITAL CHARACTERISTICS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

MAXIMUM PAYLOAD WEIGHT (IN KG): XXXXXXXX

	TIME	INTENSITY	DOWNRANGE	ALTITUDE		TIME	INTENSITY	DOWNRANGE	ALTITUDE
1:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	17:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
2:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	18:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
3:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	19:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
4:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	20:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
5:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	21:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
6:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	22:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
7:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	23:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
8:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	24:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
9:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	25:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
10:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	26:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
11:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	27:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
12:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	28:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
13:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	29:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
14:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	30:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
15:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	31:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
16:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	32:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

REMARKS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

FIGURE A-9
LAUNCH VEHICLE REVIEW OUTPUT SCREEN

```
LAUNCH SITE NAME: ------------------------------------------------------------
LAUNCH PAD NAME: -------------------------------------------------------------
LAUNCH PAD TYPE: -------------------------------------------------------------
BE NUMBER: -------------------------------------------------------------------
  LEFT HALF: -----------------------------------------------------------------
  RIGHT HALF: ----------------------------------------------------------------
LAUNCH VEHICLES: -------------------------------------------------------------
PAYLOAD MISSIONS: ------------------------------------------------------------
```

FIGURE A-10
LAUNCH SITE REVIEW INPUT SCREEN

```
                                                              XXXX / XXXX
LAUNCH SITE NAME  XXXXXXXX
LAUNCH PAD NAME:  XXXXXXXX   LAUNCH PAD TYPE(SPACE OR MISSILE): XXXXXXXX
B.E. NUMBER XXXX - XXXX
SITE LOCATION: LATITUDE:  XXXXXXXX
               LONGITUDE: XXXXXXXX
               ALTITUDE:  XXXXXXXX
LAUNCH VEHICLE CAPABILITIES OF THIS SITE:
  (1): XXXXXXXX
  (2): XXXXXXXX
  (3): XXXXXXXX
  (4): XXXXXXXX
  (5): XXXXXXXX
MISSIONS CAPABLE OF BEING LAUNCHED FROM THIS SITE:
  (1): XXXXXXXX
  (2): XXXXXXXX
  (3): XXXXXXXX
  (4): XXXXXXXX
  (5): XXXXXXXX
```

FIGURE A-11
LAUNCH SITE REVIEW OUTPUT SCREEN

Blue Ground Based Sensor System Review

Pressing the BLUE RADAR SYSTEMS soft key initiates the "Blue radar" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-12. The analyst may search on any combination of:

Sensor Name

Sensor Type

Sensor SDC Number

After the search criteria are sent to the application, all Blue ground based sensor systems are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-13. The remaining records may be observed by initiating the Next, Current or Previous review functions.

Blue Spaceborne Sensor System Review

Pressing the BLUE SPACEBORNE SYSTEMS soft key initiates the "Blue spaceborne sensor" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-14. The analyst may search on any combination of:

Sensor Identification Number

Sensor Name

After the search criteria are sent to the application, all Blue spaceborne sensor systems are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-15. The remaining records may be observed by initiating the Next, Current or Previous review functions.

```
SENSOR NAME: ------------------------------------------------------------
SENSOR TYPE: ------------------------------------------------------------
SENSOR SDC NUMBER: -----------------------------------------------------
```

FIGURE A-12
BLUE GROUND BASED SENSOR SYSTEM REVIEW INPUT SCREEN

```
SENSOR NAME: XXXXXXXX                                              XXXX / XXXX
SENSOR TYPE: XXXXXXX
SENSOR SDC NUMBER: XXXXXXXX
SENSOR LOCATION: LATITUDE: XXXXX    LONGITUDE: XXXXX    ALTITUDE: XXXXX
RANGE FIELD: XXXXXXXX
AZIMUTH MINIMUM: XXXXXXXX
AZIMUTH MAXIMUM: XXXXXXXX
ELEVATION MINIMUM: XXXXXXXX
ELEVATION MAXIMUM: XXXXXXXX
```

FIGURE A-13
BLUE GROUND BASED SENSOR SYSTEM REVIEW OUTPUT SCREEN

```
SENSOR ID NUMBER: ----------------------------------------------------------
SENSOR NAME: ---------------------------------------------------------------
```

FIGURE A-14
BLUE SPACEBORNE SENSOR SYSTEM REVIEW INPUT SCREEN

```
SENSOR ID NUMBER: XXXXXXXX                                       XXXX  / XXXX
SENSOR NAME: XXXXXXXX
SENSOR ORBIT: YEAR: XXXXXXXX
              EPOCH: DAY: XXX  HOUR: XX  MINUTE: XX  SECOND: XXXXXXXX
              RIGHT ASCENSION: XXXXXXXX
              ECCENTRICITY: XXXXXXXX
              INCLINATION: XXXXXXXX
              ARGUMENT OF PERIGEE: XXXXXXXX
              MEAN ANOMALY: XXXXXXXX
              MEAN MOTION: XXXXXXXX
```

FIGURE A-15
BLUE SPACEBORNE SENSOR SYSTEM REVIEW OUTPUT SCREEN

Soviet ESV Status Review

Pressing the SOVIET SOB soft key initiates the "Soviet ESV status" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-16. The analyst may search on any combination of:

Payload Identification Number

Sputnik Number

Series/Number

SPADAT Number

Payload Number

Associated Launch Number

After the search criteria are sent to the application, all Soviet ESV's are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-17. The remaining records may be observed by initiating the Next, Current or Previous review functions.

Tracking and Receiving Support Facilities Review

Pressing the RED SUPPORT FACILITIES soft key initiates the "tracking facilities" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-18. The analyst may search on any combination of:

Facility Name

Facility Type

Facility BE Number

After the search criteria are sent to the application, all tracking and

```
PAYLOAD ID: ------------------------------------------------------------
SPUTNIK NUMBER: --------------------------------------------------------
SERIES/NUMBER:  --------------------------------------------------------
SPADAT NUMBER:  --------------------------------------------------------
PAYLOAD MISSION:--------------------------------------------------------
LAUNCH ID: -------------------------------------------------------------
```

FIGURE A-16
SOVIET SPACE ORDER OF BATTLE REVIEW INPUT SCREEN

```
PAYLOAD IDENTIFICATION NUMBER XXXXXXXX                        XXXX / XXXX
SPUTNIK NUMBER XXXXXXXX
SERIES-NUMBER XXXXXXXX
SPADAT NUMBER XXXXXXXX
PAYLOAD MISSION XXXXXXXX
ASSOCIATED LAUNCH I.D. XXXXXXXX
LAUNCH SITE XXXXXXXX
LAUNCH PAD XXXXXXXX
LAUNCH DATE: MONTH XX DAY XX YEAR XXXX
LAUNCH TIME: HOUR XX MINUTE XX SECOND XXXXXXXX
PAYLOAD LIFE EXPECTANCY: MONTH XX DAY XX YEAR XXXX
ESTIMATED PAYLOAD WEIGHT XXXXXXXX
REMARKS: XXXXXXXX
```

FIGURE A-17
SOVIET SPACE ORDER OF BATTLE REVIEW OUTPUT SCREEN

```
FACILITY NAME: ------------------------------------------------------------
FACILITY TYPE: ------------------------------------------------------------
BE NUMBER:
  LEFT HALF: --------------------------------------------------------------
  RIGHT HALF: -------------------------------------------------------------
```

FIGURE A-18
TRACKING AND RECEIVING SUPPORT FACILITIES REVIEW INPUT SCREEN

receiving support facilities are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-19. The remaining records may be observed by initiating the Next, Current or Previous review functions.

Ground Based Sensor Inputs Review

Pressing the RADAR INPUTS soft key initiates the "ground based sensor inputs" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-20. The analyst may search on any combination of:

SDC Number of Observing Sensor

Identification of Object Being Observed

Object Type

Associated Launch Identification Number

After the search criteria are sent to the application, all ground based sensor inputs are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-21. The remaining records may be observed by initiating the Next, Current or Previous review functions.

IR Inputs Review

Pressing the IR SENSOR INPUTS soft key initiates the "IR inputs" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-22. The analyst may search on any combination of:

```
FACILITY NAME XXXXXXXX                                  XXXX / XXXX'
FACILITY TYPE XXXXXXXX
B.E. NUMBER XXXX - XXXX
FACILITY LOCATION: LATITUDE: XXXXXXXX
                   LONGITUDE: XXXXXXXX
                   ALTITUDE: XXXXXXXX
CHARACTERISTICS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

FIGURE A-19
TRACKING AND RECEIVING SUPPORT FACILITIES REVIEW OUTPUT SCREEN

```
SDC NUMBER: ----------------------------------------------------------------
OBJECT BEING OBSERVED: -----------------------------------------------------
OBJECT TYPE: ---------------------------------------------------------------
ASSOCIATED LAUNCH ID: ------------------------------------------------------
```

FIGURE A-20
GROUND BASED SENSOR INPUTS REVIEW INPUTS SCREEN

```
                                                                        XXXX / XXXX
OBSERVING SENSORS SDC NUMBER: XXXXXXXX
I.D. NUMBER OF OBJECT BEING OBSERVED XXXXXXXX
OBJECT TYPE XXXXXXXX
ASSOCIATED LAUNCH I.D. NUMBER XXXXXXXX
SENSOR OBSERVATIONS: YEAR: XXXX
                     EPOCH: DAY XXX HOUR XX  MINUTE XX  SECOND XXXXXXXX
                     ASCENSION: XXXXXXXX
                     ECCENTRICITY: XXXXXXXX
                     INCLINATION: XXXXXXXX
                     ARGUMENT OF PERIGEE: XXXXXXXX
                     MEAN ANOMALY: XXXXXXXX
                     MEAN MOTION: XXXXXXXX
                     N DOT / 2 : XXXXXXXX
                     N DOUBLE DOT / 6: XXXXXXXX
```

FIGURE A-21
GROUND BASED SENSOR INPUTS REVIEW OUTPUT SCREEN

```
SENSOR ID: ---------------------------------------------------------------------
SENSOR NAME:-------------------------------------------------------------------
LAUNCH ID: --------------------------------------------------------------------
```

FIGURE A-22
IR INPUTS REVIEW INPUT SCREEN

Sensor Identification Number

Sensor Name

Launch Identification Number

After the search criteria are sent to the application, all IR inputs are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-23. The remaining records may be observed by initiating the Next, Current or Previous review functions.

Polynomial Inputs Review

Pressing the POLYNOMIAL INPUTS soft key initiates the "polynomial inputs" data base review. When this option is selected, the analyst is presented with the transaction screen depicted in Figure A-24. The analyst may search on any combination of:

Sensor Identification Number

Sensor Name

Launch Identification Number

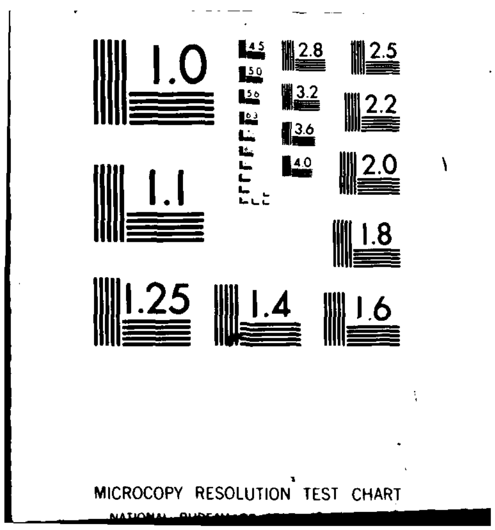
After the search criterion are sent to the application, all polynomial inputs are searched for the records matching the entered criterion. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-25. The remaining records may be observed by initiating the Next, Current or Previous review functions.

A.3.2.3 Next Review

Pressing the EXAMINE NEXT RECORD soft key initiates the Next review function. The Next review function allows the analyst to examine the next record retrieved by the last data base review function. There are no inputs to this function. The function determines what the last data base and record

AD-A110 272 PAR TECHNOLOGY CORP ROME NY F/G 9/2
SABERS. STAND-ALONE ADIC BINARY EXPLOITATION RESOURCES SYSTEM. --ETC(U)
SEP 81 A J FRANKLIN, R L CALDWELL, S COLE F30602-78-C-0078
UNCLASSIFIED RADC-TR-81-250-VOL-2 NL

2 of 4


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART

```
SENSOR IDENTIFICATION NUMBER xxxxxxxx SENSOR NAME xxxxxxxx PAGE 1:2 xxxx/xxxx
LAUNCH IDENTIFICATION NUMBER xxxxxxxx
LAUNCH DATE: MONTH xx DAY xx YEAR xxxx LAUNCH TIME:HOUR xx MIN xx SEC xxxxxxxx
LAUNCH LOCATION: LATITUDE  xxxx LONGITUDE xxxxx ALTITUDE xxxxx
SENSOR OBSERVATIONS: TIME-(HHMMSS)  INTENSITY  AZIMUTH-(RAD)  ELEVATION-(RAD)
               (1):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
               (2):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
               (3):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
               (4):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
               (5):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
               (6):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
               (7):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
               (8):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
               (9):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (10):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (11):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (12):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (13):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (14):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (15):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (16):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (17):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (18):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (19):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
                                               PAGE 2:2
SENSOR OBSERVATIONS: TIME-(HHMMSS)  INTENSITY  AZIMUTH-(RAD)  ELEVATION-(RAD)
              (20):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (21):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (22):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (23):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (24):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (25):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (26):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
              (27):  XXXXXXXX      XXXXXXXX   XXXXXXXX       XXXXXXXX
```

Figure A-23 IR Inputs Review Output Screen

```
SENSOR ID: ------------------------------------------------------------------
SENSOR NAME: ----------------------------------------------------------------
LAUNCH ID: ------------------------------------------------------------------
```

FIGURE A-24
POLYNOMIAL INPUTS REVIEW INPUT SCREEN

```
SENSOR IDENTIFICATION NUMBERXXXXXXXX SENSOR NAME XXXXXXXX PAGE 1:2 XXXX/XXXX
LAUNCH IDENTIFICATION NUMBER XXXXXXXX
LAUNCH DATE: MONTH XX DAY XX YEAR XXXX LAUNCH TIME:HOUR XX MIN XX SEC XXXXXXXX
LAUNCH LOCATION: LATITUDE XXXXX LONGITUDE XXXXX ALTITUDE XXXXX
SENSOR OBSERVATIONS:
            HH MM SS                      SECONDS
1)     TIME             INTERVAL LENGTH: XXXXXXXX

X(0)= XXXXXXXX   X(1)= XXXXXXXX   X(2)= XXXXXXXX   X(3)= XXXXXXXX
Y(0)= XXXXXXXX   Y(1)= XXXXXXXX   Y(2)= XXXXXXXX   Y(3)= XXXXXXXX
Z(0)= XXXXXXXX   Z(1)= XXXXXXXX   Z(2)= XXXXXXXX   Z(3)= XXXXXXXX

2)     TIME             INTERVAL LENGTH: XXXXXXXX

X(0)= XXXXXXXX   X(1)= XXXXXXXX   X(2)= XXXXXXXX   X(3)= XXXXXXXX
Y(0)= XXXXXXXX   Y(1)= XXXXXXXX   Y(2)= XXXXXXXX   Y(3)= XXXXXXXX
Z(0)= XXXXXXXX   Z(1)= XXXXXXXX   Z(2)= XXXXXXXX   Z(3)= XXXXXXXX

3)     TIME             INTERVAL LENGTH: XXXXXXXX

X(0)= XXXXXXXX   X(1)= XXXXXXXX   X(2)= XXXXXXXX   X(3)= XXXXXXXX
Y(0)= XXXXXXXX   Y(1)= XXXXXXXX   Y(2)= XXXXXXXX   Y(3)= XXXXXXXX
Z(0)= XXXXXXXX   Z(1)= XXXXXXXX   Z(2)= XXXXXXXX   Z(3)= XXXXXXXX

                                                        PAGE 2:2
            HH MM SS                      SECONDS
4)     TIME             INTERVAL LENGTH: XXXXXXXX

X(0)= XXXXXXXX   X(1)= XXXXXXXX   X(2)= XXXXXXXX   X(3)= XXXXXXXX
Y(0)= XXXXXXXX   Y(1)= XXXXXXXX   Y(2)= XXXXXXXX   Y(3)= XXXXXXXX
Z(0)= XXXXXXXX   Z(1)= XXXXXXXX   Z(2)= XXXXXXXX   Z(3)= XXXXXXXX

5)     TIME             INTERVAL LENGTH: XXXXXXXX

X(0)= XXXXXXXX   X(1)= XXXXXXXX   X(2)= XXXXXXXX   X(3)= XXXXXXXX
Y(0)= XXXXXXXX   Y(1)= XXXXXXXX   Y(2)= XXXXXXXX   Y(3)= XXXXXXXX
Z(0)= XXXXXXXX   Z(1)= XXXXXXXX   Z(2)= XXXXXXXX   Z(3)= XXXXXXXX
```

Figure A-25 Polynomial Inputs Review Output Screen

reviewed were. The output information screen presented depends on the last data base reviewed.

If the last data base reviewed was	The output screen is Figure
"launch folder"	A-7
"launch vehicle"	A-9
"launch site"	A-11
"Blue radar"	A-13
"Blue spaceborne sensor"	A-15
"Soviet ESV status"	A-17
"tracking facilities"	A-19
"ground based sensor inputs"	A-21
"IR inputs"	A-23
"polynomial inputs"	A-25

A.3.2.4 Current Review

Pressing the EXAMINE CURRENT RECORD soft key initiates the Current review function. The Current review function allows the analyst to reexamine the current record retrieved by the last data base review function. There are no inputs to this function. The function determines what the last data base and record reviewed were. The output information screen presented depends on the last data base reviewed.

If the last data base reviewed was	The output screen is Figure
"launch folder"	A-7
"launch vehicle"	A-9
"launch site"	A-11
"Blue radar"	A-13
"Blue spaceborne sensor"	A-15
"Soviet ESV status"	A-17
"tracking facilities"	A-19
"ground based sensor inputs"	A-21
"IR inputs"	A-23
"polynomial inputs"	A-25

A.3.2.5 Previous Review

Pressing the EXAMINE PREVIOUS RECORD soft key initiates the Previous review function. The Previous review function allows the analyst to examine the record previous to the current record retrieved by the last data base function. There are no inputs to this function. The function determines what the last data base and record reviewed were. The output information screen presented depends on the last data base reviewed.

If the last data base reviewed was	The output screen is Figure
"launch folder"	A-7
"launch vehicle"	A-9
"launch site"	A-11
"Blue radar"	A-13
"Blue spaceborne sensor"	A-15
"Soviet ESV status"	A-17
"tracking facilities"	A-19
"ground based sensor inputs"	A-21
"IR inputs"	A-23
"polynomial inputs"	A-25

A.3.2.6 Current Launch Review

The Current Launch Review function is designed to assist the analyst in comparing the current launch event summary information contained in the current launch folder with historical launch events. The application extracts the information contained in the current launch folder record and prepares the transaction screen depicted in Figure A-26 by formatting default information into every field but the "LAUNCH ID:" field. After the search criteria is sent to the application, all launch folders are searched for the records matching the entered criteria. Once the records are retrieved, the first record is displayed to the analyst as depicted in Figure A-7. The remaining records may be observed by initiating the Next, Current or Previous review functions.

Pressing the CURRENT LAUNCH REVIEW soft key initiates the Current Launch Review function. The current launch folder is retrieved, and the contents of the record are examined. If the launch site and/or the launch site pad are known, these names are entered into the "LAUNCH SITE:" and/or "LAUNCH PAD:"

```
LAUNCH ID: ------------------------------------------------------------
LAUNCH DATE: MONTH: ---------------------------------------------------
             DAY: -----------------------------------------------------
             YEAR: ----------------------------------------------------
LAUNCH TIME: HOUR: ----------------------------------------------------
             MINUTE: --------------------------------------------------
             SECOND: --------------------------------------------------
LAUNCH POSITION: LAT: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 LON: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 ALT: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LAUNCH SITE: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LAUNCH PAD:  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LAUNCH VEHICLE: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LAUNCH AZIMUTH: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LAUNCH INCLINATION: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
EVENT TYPE (SPACE,MISSILE): XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
THREAT OR NOTHREAT: ---------------------------------------------------
PAYLOAD MISSION: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
TARGET SATELLITE ID(FOR ASAT ONLY): -----------------------------------
LAUNCHED SATELLITE ID: ------------------------------------------------
```

FIGURE A-26
CURRENT LAUNCH REVIEW INPUT SCREEN

fields, and the launch position is entered into the "LAUNCH POSITION: LAT:" and "LON:" fields.

Otherwise, if the launch position is given, a range assertion is created for the "LAT:" and "LON:" fields in which the minimum latitude (longitude) is the given latitude (longitude) less one degree, and the maximum latitude (longitude) is the given latitude (longitude) plus one degree. Information given in the current launch folder for the launch vehicle, event type and payload mission are entered into the appropriate fields. Range assertions are generated for the launch azimuth and the launch inclination by subtracting 2.5 degrees from the given value for the minimum angle and adding 2.5 degrees to the given value for the maximum angle. The field on the transaction screen is left blank if the corresponding information is not present in the current launch folder record. If the current launch folder record is empty, a blank transaction screen is presented to the analyst (similar to the launch folder review function).

A.3.2.7 Summary Review

The purpose of the Summary review function is to provide a line printer listing of all the records in a data base which satisfy the search criteria entered by the analyst. This allows the analyst to view all the information contained in each of these records at one time. Additional flexibility is provided for the analyst because he is also able to specify which fields in the data base are to be listed.

The Summary review function is unlike any of the other review functions in that the "last review" system data base is not altered by the system after exercising this function. This means that running this application does not reset the last data base reviewed information. A further difference is that the "polynomial inputs" data base is not available for summarizing.

Pressing the SUMMARY soft key initiates the Summary review function. The analyst is asked to enter the identity of the data base he wishes to summarize. The transaction screen is depicted in Figure A-27. The last data base reviewed is presented as the default data base to summarize. After the data base is selected, a blank transaction screen is presented to the analyst. The analyst enters the search criteria in the same manner as for a data base review function. The transaction screen presented depends on the data base to be summarized.

If the data base to be summarized is	The input screen is Figure
"launch folder"	A-6
"launch vehicle"	A-8
"launch site"	A-10
"Blue radar"	A-12
"Blue spaceborne sensor"	A-14
"Soviet ESV status"	A-16
"tracking facilities"	A-18
"ground based sensor inputs"	A-20
"IR inputs"	A-22

If any records are retrieved because they match the search criteria, the analyst is presented with a third screen upon which the analyst indicates the fields he wishes to be listed by putting an asterix in the appropriate field. Again, the screen depends on which data base is being summarized.

```
WHICH DATA BASE IS TO BE SUMMARIZED
 1 - LAUNCH FOLDER
 2 - IR INPUTS
 3 - LAUNCH VEHICLES
 4 - BLUE SPACEBORNE SENSOR SYSTEMS
 5 - LAUNCH SITE FILE
 6 - TRACKING AND RECIEVING SUPPORT FACILITIES
 7 - BLUE GROUND BASED SENSOR SYSTEMS
 8 - SOVIET ESV STATUS FILE
 9 - GROUND BASED SENSOR INPUTS

ENTER NUMBER OF THE DATA BASE TO BE SUMMARIZED XX
```

FIGURE A-27
SUMMARY INPUT SCREEN

If the data base to be summarized is	The input screen is Figure
"launch folder"	A-28
"launch vehicle"	A-29
"launch site"	A-30
"Blue radar"	A-31
"Blue spaceborne sensor"	A-32
"Soviet ESV status"	A-33
"tracking facilities"	A-34
"ground based sensor inputs"	A-35
"IR inputs"	A-36

The output of the application is a listing automatically printed on the line printer. However, the file is not suitable for typing on the monitor since the line width of the listing is greater than the line width of the monitor. The listing consists of the header, which lists the name of the data base and the names of the fields for which the analyst entered assertions followed by the assertions, and the information presented in a matrix format. The name of each requested field is printed as the column heading, and the values of the different records are listed in the column. A sample output is presented in Figure A-37.

```
LAUNCH IDENTIFICATION NUMBER: -
PRE-LAUNCH INFORMATION: -
LAUNCH DATE: -
LAUNCH TIME: _
LAUNCH POSITION: -
LAUNCH SITE: _            LAUNCH PAD: -
CONFIRMATION SOURCES: -
LAUNCH VEHICLE: _
LAUNCH AZIMUTH: _       LAUNCH INCLINATION: _
REENTRY LOCATION: _
REENTRY DATE: -
REENTRY TIME: -
REENTRY AZIMUTH: -      REENTRY INCLINATION: _
REENTRY CONFIRMATION SOURCE: -
EVENT TYPE (SPACE, MISSILE): _
THREAT OR NOTHREAT: -
PAYLOAD MISSION: _
TARGET SATELLITE ID (FOR ASAT ONLY): _
LAUNCHED SATELLITE ID: -
OTHER OBJECTS: -
REMARKS: -

PLACE A * IN ANY FIELD WHICH YOU WISH TO SUMMARIZE.
```

FIGURE A-28
LAUNCH INPUT SCREEN FOLDER SUMMARY

```
LAUNCH VEHICLE: -
PAYLOAD MISSION: -
ORBITAL CHARACTERISTICS: -
MAXIMUM PAYLOAD WEIGHT (IN KG): -
TIME: -
INTENSITY: _
DOWNRANGE: -
ALTITUDE: -
REMARKS: -

PLACE A * IN ALL FIELDS THAT YOU WISH TO HAVE SUMMARIZED.
```

FIGURE A-29
LAUNCH VEHICLE SUMMARY INPUT SCREEN

```
LAUNCH SITE NAME: -
LAUNCH PAD NAME: -
LAUNCH PAD TYPE(SPACE OR MISSILE): -
B.E. NUMBER: -
SITE LOCATION: -
LAUNCH VEHICLE CAPABILITIES OF THIS SITE: -
MISSIONS CAPABLE OF BEING LAUNCHED FROM THIS SITE: -

PLACE A * IN ALL FIELDS THAT YOU WISH TO SUMMARIZE.
```

FIGURE A-30
LAUNCH SITE SUMMARY INPUT SCREEN

```
SENSOR NAME: -
SENSOR TYPE: -
SENSOR SDC NUMBER: _
SENSOR LOCATION: _
RANGE FIELD: -
AZIMUTH: -
ELEVATION: -

PLACE A * IN ALL FIELDS THAT YOU WISH TO SUMMARIZE.
```

FIGURE A-31
BLUE GROUND BASED SENSOR SYSTEM SUMMARY INPUT SCREEN

```
SENSOR ID NUMBER: -
SENSOR NAME: -
SENSOR ORBIT: YEAR: -
              EPOCH: -
              RIGHT ASCENSION: -
              ECCENTRICITY: -
              INCLINATION: -
              ARGUMENT OF PERIGEE: -
              MEAN ANOMALY: -
              MEAN MOTION: -

PLACE A * IN ALL FIELDS THAT YOU WISH TO SUMMARIZE.
```

FIGURE A-32
BLUE SPACEBORNE SENSOR SYSTEM SUMMARY INPUT SCREEN

```
PAYLOAD IDENTIFICATION NUMBER: -
SPUTNIK NUMBER: -
SERIES-NUMBER: -
SPADAT NUMBER: _
PAYLOAD MISSION: _
ASSOCIATED LAUNCH I.D.: _
LAUNCH SITE:  -
LAUNCH PAD: -
LAUNCH DATE: _
LAUNCH TIME: _
PAYLOAD LIFE EXPECTANCY: -
ESTIMATED PAYLOAD WEIGHT: -
REMARKS: _

PLACE A * IN ALL FIELDS THAT YOU WISH TO SUMMARIZE.
```

FIGURE A-33
SOVIET SPACE ORDER OF BATTLE SUMMARY INPUT SCREEN

```
FACILITY NAME: -
FACILITY TYPE: -
B.E. NUMBER: -
FACILITY LOCATION: -
CHARACTERISTICS: -

PLACE A * IN ALL FIELDS THAT YOU WISH TO SUMMARIZE.
```

FIGURE A-34
TRACKING AND RECEIVING SUPPORT FACILITIES SUMMARY INPUT SCREEN

```
OBSERVING SENSORS SDC NUMBER: -
I.D. NUMBER OF OBJECT BEING OBSERVED -
OBJECT TYPE -
ASSOCIATED LAUNCH I.D. NUMBER -
SENSOR OBSERVATIONS: YEAR: -
                     EPOCH: -
                     ASCENSION: -
                     ECCENTRICITY: -
                     INCLINATION: -
                     ARGUMENT OF PERIGEE: -
                     MEAN ANOMALY: -
                     MEAN MOTION: -
                     N DOT / 2 : -
                     N DOUBLE DOT / 6: -

PLACE A * IN ALL FIELDS THAT YOU WISH TO HAVE SUMMARIZED.
```

FIGURE A-35
GROUND BASED SENSOR INPUTS SUMMARY INPUT SCREEN

```
SENSOR IDENTIFICATION NUMBER: -
SENSOR NAME: -
LAUNCH IDENTIFICATION NUMBER -
LAUNCH DATE: -
LAUNCH TIME: -
LAUNCH LOCATION: -
SENSOR OBSERVATIONS: TIME: -
                     INTENSITY: -
                     AZIMUTH: -
                     ELEVATION: -

PLACE A * IN ALL FIELDS THAT YOU WISH TO HAVE SUMMARIZED.
```

FIGURE A-36
IR INPUTS SUMMARY INPUT SCREEN

```
*******************************************************************************************
* THIS TABLE SUMMARIZES THOSE RECORDS RETRIEVED FROM THE LAUNCHSITE FILE                  *
* BASED UPON THE FOLLOWING ASSERTIONS:                                                    *
* 1ST HALF OF BE NUMBER: 21,22                                                            *
* 2ND HALF OF BE NUMBER: (1,100)                                                          *
*******************************************************************************************
```

	LAUNCH SITE	LAUNCH PAD NAME	TYPE	BE NUMBER	SITE LOCATION LATITUDE	LONGITUDE	ALTITUDE
[1]	ASITE	PAD1	MISSILE	21- 1	48.26	5.50	
[2]	ASITE	P2	SPACE	21- 2	48.38	5.45	
[3]	ASITE	PA3	MISSILE	21- [illegible]	48.22	5.55	
[4]	BSITE	PAD1	BOTH	22- 1	52.00	0.10	
[5]	CSITE	PAD1	MISSILE	22- 5	15.00	10.00	
[6]	ASITE	PAD4	SPACE	21- 5	48.44	5.55	
[7]	ASITE	P5	MISSILE	21- 7	48.50	5.50	
[8]	ASITE	PA6	BOTH	21- 10	48.15	5.45	
[9]	ASITE	PAD7	SPACE	21- 3	48.25	5.50	

	CAPABILITY(1)	CAPABILITY(2)	CAPABILITY(3)	CAPABILITY(4)	CAPABILITY(5)
[1]	VEHICLE1				
[2]	VEHICLE2				
[3]	VEHICLE1				
[4]	VEHICLE1	VEHICLE2			
[5]	VEHICLE1				
[6]	VEHICLE3				
[7]	VEHCILE1				
[8]	VEHICLE2	VEHICLE1			
[9]	VEHICLE3	VEHICLE2			

	MISSION(1)	MISSION(2)	MISSION(3)	MISSION(4)	MISSION(5)
[1]					
[2]	MISSION1	MISSION3			
[3]					
[4]	MISSION1				
[5]					
[6]	MISSION2				
[7]					
[8]					
[9]	MISSION1	MISSION3	MISSION2	MISSION4	

FIGURE A-37
EXAMPLE OF SUMMARY OUTPUT

A.3.3 Data Base Update Functions

The analyst may be required to add information about previously unknown capabilities or to correct inaccurate information for an existing data base record. The update functions allow the analyst to make these modifications. In addition, the Select Launch ID and Select Payload ID functions also cause the system to change the current launch event identification number.

A.3.3.1 Data Base Update

Pressing the UPDATE AN EXISTING RECORD soft key initiates the update data base application. When this application is selected, the analyst is presented with an initial transaction screen which depends on the last data base reviewed.

If the last data base reviewed was	The input screen is Figure
"launch folder"	A-38
"launch vehicle"	A-39
"launch site"	A-40
"Blue radar"	A-41
"Blue spaceborne sensor"	A-42
"Soviet ESV status"	A-43
"tracking facilities"	A-44
"ground based sensor inputs"	A-45
"IR inputs"	A-46
"polynomial inputs"	A-47

At the top of the transaction screen, the analyst is informed what the last data base reviewed was. At the bottom of the screen, the application outputs the unique characteristics of the last record that was reviewed. This application is designed with the philosophy that the analyst will most likely

```
THE DATA BASE TO USE TO PERFORM THIS OPERATION IS: XX

 WHERE:
 1 IS THE LAUNCH FOLDER FILE.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECIEVING SUPPORT FACILITEIS FILE.
 7 IS THE BLUE GROUND BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND BASED SENSOR INPUTS FILE.
10 IS THE POLYNOMIAL FILE.
THE LAUNCH ID TO USE IN THIS OPERATION IS: XXXXXXXX
```

FIGURE A-38
LAUNCH FOLDER UPDATE AND DELETE INPUT SCREEN

```
THE DATA BASE TO USE TO PERFORM THIS OPERATION IS: XX

 WHERE:
 1 IS THE LAUNCH FOLDER FILE.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECIEVING SUPPORT FACILITEIS FILE.
 7 IS THE BLUE GROUND BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND BASED SENSOR INPUTS FILE.
10 IS THE POLYNOMIAL FILE.
THE LAUNCH VEHICLE TO BE USED IS: XXXXXXXX
THE CORRESPONDING PAYLOAD MISSION IS: XXXXXXXX
```

FIGURE A-39
LAUNCH VEHICLE UPDATE AND DELETE INPUT SCREEN

```
THE DATABASE TO USE TO PERFORM THIS OPERATION IS: XX

WHERE:
 1 IS THE LAUNCH FOLDER.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECEIVING SUPPORT FACILITIES FILE.
 7 IS THE BLUE GROUND-BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND-BASED SENSOR INPUTS DATA FILE.
10 IS THE POLYNOMIAL FILE.
THE B.E. NUMBER TO BE USED IN THIS OPERATION IS: XXXX - XXXX
```

FIGURE A-40
LAUNCH SITE UPDATE AND DELETE INPUT SCREEN

```
THE DATABASE TO USE TO PERFORM THIS OPERATION IS: XX

WHERE:
 1 IS THE LAUNCH FOLDER.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECEIVING SUPPORT FACILITIES FILE.
 7 IS THE BLUE GROUND-BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND-BASED SENSOR INPUTS DATA FILE.
10 IS THE POLYNOMIAL FILE.

THE SDC NUMBER TO BE USED IN THIS OPERATION IS: XXXXXXXX
```

FIGURE A-41
BLUE GROUND BASED SENSOR SYSTEM UPDATE AND DELETE INPUT SCREEN

```
THE DATA BASE TO USE TO PERFORM THIS OPERATION IS: XX

 WHERE:
 1 IS THE LAUNCH FOLDER FILE.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECIEVING SUPPORT FACILITEIS FILE.
 7 IS THE BLUE GROUND BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND BASED SENSOR INPUTS FILE.
10 IS THE POLYNOMIAL FILE.
THE SENSOR ID TO BE USED IN THIS OPERATION IS: XXXXXXXX
```

FIGURE A-42
BLUE SPACEBORNE SENSOR SYSTEM UPDATE AND DELETE INPUT SCREEN

```
THE DATABASE TO USE TO PERFORM THIS OPERATION IS: XX

WHERE:
 1 IS THE LAUNCH FOLDER.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECEIVING SUPPORT FACILITIES FILE.
 7 IS THE BLUE GROUND-BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND-BASED SENSOR INPUTS DATA FILE.
10 IS THE POLYNOMIAL FILE.

THE PAYLOAD ID TO BE USED IN THIS OPERATION IS: XXXXXXXX
```

FIGURE A-43
SOVIET SPACE ORDER OF BATTLE UPDATE AND DELETE INPUT SCREEN

```
THE DATABASE TO USE TO PERFORM THIS OPERATION IS: XX

WHERE:
 1 IS THE LAUNCH FOLDER.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECEIVING SUPPORT FACILITIES FILE.
 7 IS THE BLUE GROUND-BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND-BASED SENSOR INPUTS DATA FILE.
10 IS THE POLYNOMIAL FILE.

THE B.E. NUMBER TO BE USED IN THIS OPERATION IS: XXXX - XXXX
```

FIGURE A-44
TRACKING AND RECEIVING SUPPORT FACILITIES UPDATE AND DELETE INPUT SCREEN

```
THE DATABASE TO USE TO PERFORM THIS OPERATION IS:  XX

WHERE:
 1 IS THE LAUNCH FOLDER.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECEIVING SUPPORT FACILITIES FILE.
 7 IS THE BLUE GROUND-BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND-BASED SENSOR INPUTS DATA FILE.
10 IS THE POLYNOMIAL FILE.

THE ID OF THE OBSERVED OBJECT IS: XXXXXXXX
THE EPOCH YEAR IS: XXXX
THE EPOCH DAY IS:  XX
THE EPOCH HOUR IS: XX
THE EPOCH MINUTE IS: XX
```

FIGURE A-45
GROUND BASED SENSOR INPUTS UPDATE AND DELETE INPUT SCREEN

```
THE DATA BASE TO USE TO PERFORM THIS OPERATION IS: XX

 WHERE:
 1 IS THE LAUNCH FOLDER FILE.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECIEVING SUPPORT FACILITEIS FILE.
 7 IS THE BLUE GROUND BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND BASED SENSOR INPUTS FILE.
10 IS THE POLYNOMIAL FILE.
THE SENSOR ID TO BE USED IN THIS OPERATION IS: XXXXXXXX
THE CORRESPONDING LAUNCH ID TO USE IS: XXXXXXXX
```

FIGURE A-46
IR INPUTS UPDATE AND DELETE INPUT SCREEN

```
THE DATA BASE TO USE TO PERFORM THIS OPERATION IS: XX

 WHERE:
 1 IS THE LAUNCH FOLDER FILE.
 2 IS THE IR INPUTS FILE.
 3 IS THE LAUNCH VEHICLES FILE.
 4 IS THE BLUE SPACEBORNE SENSOR SYSTEMS FILE.
 5 IS THE LAUNCH SITE FILE.
 6 IS THE TRACKING AND RECIEVING SUPPORT FACILITE'S FILE.
 7 IS THE BLUE GROUND BASED SENSOR SYSTEMS FILE.
 8 IS THE SOVIET ESV STATUS FILE.
 9 IS THE GROUND BASED SENSOR INPUTS FILE.
10 IS THE POLYNOMIAL FILE.
THE SENSOR ID TO BE USED IN THIS OPERATION IS: XXXXXXXX
THE CORRESPONDING LAUNCH ID TO USE IS: XXXXXXXX
```

FIGURE A-47
POLYNOMIAL INPUTS UPDATE AND DELETE INPUT SCREEN

want to update the last record he reviewed.

If the analyst wishes to update a record other than the one which was last reviewed, he merely has to change the unique characteristics shown at the bottom of the screen. If the analyst wishes to update a record in some other data base, he has to modify the data base identifier at the top of the screen. If the data base identifier is modified, a new transaction screen is presented to the analyst, according to the data base to be updated.

If the data base to be updated is	The input screen is Figure
"launch folder"	A-38
"launch vehicle"	A-39
"launch site"	A-40
"Blue radar"	A-41
"Blue spaceborne sensor"	A-42
"Soviet ESV status"	A-43
"tracking facilities"	A-44
"ground based sensor inputs"	A-45
"IR inputs"	A-46
"polynomial inputs"	A-47

The data base identifier at the top of the screen indicates which data base is to be updated. However, since the record to be updated is not the last record to be reviewed, no characteristics are presented to the analyst as defaults at the bottom of the screen. The analyst must enter the unique characteristics of the record to be updated.

Once the data base and the record of the data base have been identified, the analyst is presented with a screen upon which he makes his changes. All the information stored in the data base for that record is presented as default information on the screen. The screen presented to the analyst for

adding and modifying the information depends on the data base being updated.

If the data base to be updated is	The modify screen is Figure
"launch folder"	A-48
"launch vehicle"	A-49
"launch site"	A-50
"Blue radar"	A-51
"Blue spaceborne sensor"	A-52
"Soviet ESV status"	A-53
"tracking facilities"	A-54
"ground based sensor inputs"	A-55
"IR inputs"	A-56
"polynomial inputs"	A-57

A.3.3.2 Launch Event Update

As described in Section A.3.1.1, the "launch folder" data base acts as the centralized data base for the analyst and the applications to link together all the information associated with each launch event. To uniquely identify one folder within the data base of launch folders requires a launch identification number. SABERS maintains the identification number of the current event in the "launch id" system data base. All the other applications within SABERS which require information associated with a launch event use this current launch identification number as the default launch identification number. The analyst causes the system to set the "launch id" data base by updating the launch folder with either the Select Launch ID application or the Select Payload ID application.

```
LAUNCH IDENTIFICATION NUMBER XXXXXXXX
PRE-LAUNCH INFORMATION XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LAUNCH DATE: MONTH XX DAY XX YEAR XXX
LAUNCH TIME: HOUR XX MINUTE XX SECOND XXXXXXXX
LAUNCH POSITION: LATITUDE: XXXXX LONGITUDE: XXXXX ALTITUDE: XXXXX
LAUNCH SITE: XXXXXXXX     LAUNCH PAD: XXXXXXXX
CONFIRMATION SOURCES: XXXXXXXX
LAUNCH VEHICLE: XXXXXXXX
LAUNCH AZIMUTH: XXXXX   LAUNCH INCLINATION: XXXXX
REENTRY LOCATION: LATITUDE: XXXXX   LONGITUDE: XXXXX
REENTRY DATE: MONTH XX DAY XX YEAR XXXX
REENTRY TIME: HOUR XX MINUTE XX SECOND XXXXXXXX
REENTRY AZIMUTH: XXXXX   REENTRY INCLINATION: XXXXX
REENTRY CONFIRMATION SOURCE: XXXXXXXX
EVENT TYPE (SPACE, MISSILE): XXXXXXXX
THREAT OR NOTHREAT: XXXXXXXX
PAYLOAD MISSION: XXXXXXXX
TARGET SATELLITE ID (FOR ASAT ONLY): XXXXXXXX
LAUNCHED SATELLITE ID: XXXXXXXX
OTHER OBJECTS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
REMARKS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

FIGURE A-48
LAUNCH FOLDER MODIFY AND ADD RECORD INPUT SCREEN

LAUNCH VEHICLE: XXXXXXXX
PAYLOAD MISSION: XXXXXXXX
ORBITAL CHARACTERISTICS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
MAXIMUM PAYLOAD WEIGHT (IN KG): XXXXXXXX

	TIME	INTENSITY	DOWNRANGE	ALTITUDE		TIME	INTENSITY	DOWNRANGE	ALTITUDE
1:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	17:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
2:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	18:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
3:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	19:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
4:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	20:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
5:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	21:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
6:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	22:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
7:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	23:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
8:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	24:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
9:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	25:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
10:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	26:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
11:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	27:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
12:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	28:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
13:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	29:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
14:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	30:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
15:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	31:	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
16:					32:				

REMARKS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

FIGURE A-49
LAUNCH VEHICLE MODIFY AND ADD RECORD INPUT SCREEN

```
LAUNCH SITE NAME XXXXXXXX
LAUNCH PAD NAME: XXXXXXXX    LAUNCH PAD TYPE(SPACE OR MISSILE): XXXXXXXX
B.E. NUMBER XXXX - XXXX
SITE LOCATION: LATITUDE:  XXXXXXXX
               LONGITUDE: XXXXXXXX
               ALTITUDE:  XXXXXXXX
LAUNCH VEHICLE CAPABILITIES OF THIS SITE:
  (1): XXXXXXXX
  (2): XXXXXXXX
  (3): XXXXXXXX
  (4): XXXXXXXX
  (5): XXXXXXXX
MISSIONS CAPABLE OF BEING LAUNCHED FROM THIS SITE:
  (1): XXXXXXXX
  (2): XXXXXXXX
  (3): XXXXXXXX
  (4): XXXXXXXX
  (5): XXXXXXXX
```

FIGURE A-50
LAUNCH SITE MODIFY AND ADD RECORD INPUT SCREEN

```
SENSOR NAME: XXXXXXXX
SENSOR TYPE: XXXXXXXX
SENSOR SDC NUMBER: XXXXXXXX
SENSOR LOCATION: LATITUDE: XXXXX   LONGITUDE: XXXXX   ALTITUDE: XXXXX
RANGE FIELD: XXXXXXXX
AZIMUTH MINIMUM: XXXXXXXX
AZIMUTH MAXIMUM: XXXXXXXX
ELEVATION MINIMUM: XXXXXXXX
ELEVATION MAXIMUM: XXXXXXXX
```

FIGURE A-51
BLUE GROUND BASED SENSOR SYSTEM MODIFY AND ADD RECORD INPUT SCREEN

```
SENSOR ID NUMBER: XXXXXXXX
SENSOR NAME: XXXXXXXX
SENSOR ORBIT: YEAR: XXXX
              EPOCH: DAY: XXX  HOUR: XX  MINUTE: XX  SECOND: XXXXXXXX
              RIGHT ASCENSION: XXXXXXXX
              ECCENTRICITY: XXXXXXXX
              INCLINATION: XXXXXXXX
              ARGUMENT OF PERIGEE: XXXXXXXX
              MEAN ANOMALY: XXXXXXXX
              MEAN MOTION: XXXXXXXX
```

FIGURE A-52
BLUE SPACEBORNE SENSOR SYSTEM MODIFY AND ADD RECORD INPUT SCREEN

```
PAYLOAD IDENTIFICATION NUMBER XXXXXXXX
SPUTNIK NUMBER XXXXXXXX
SERIES-NUMBER XXXXXXXX
SPADAT NUMBER XXXXXXXX
PAYLOAD MISSION XXXXXXXX
ASSOCIATED LAUNCH I.D. XXXXXXXX
LAUNCH SITE XXXXXXXX
LAUNCH PAD XXXXXXXX
LAUNCH DATE: MONTH XX  DAY XX  YEAR XXXX
LAUNCH TIME: HOUR XX  MINUTE XX  SECOND XXXXXXXX
PAYLOAD LIFE EXPECTANCY: MONTH XX  DAY XX  YEAR XXXX
ESTIMATED PAYLOAD WEIGHT XXXXXXXX
REMARKS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

FIGURE A-53
SOVIET SPACE ORDER OF BATTLE MODIFY AND ADD RECORD INPUT SCREEN

```
FACILITY NAME XXXXXXXX
FACILITY TYPE XXXXXXXX
B.E. NUMBER XXXX - XXXX
FACILITY LOCATION: LATITUDE: XXXXXXXX
                   LONGITUDE:XXXXXXXX
                   ALTITUDE: XXXXXXXX
CHARACTERISTICS: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

FIGURE A-54
TRACKING AND RECEIVING SUPPORT FACILITIES MODIFY AND ADD RECORD INPUT SCREEN

```
OBSERVING SENSORS SDC NUMBER: XXXXXXXX
I.D. NUMBER OF OBJECT BEING OBSERVED XXXXXXXX
OBJECT TYPE  XXXXXXXX
ASSOCIATED LAUNCH I.D. NUMBER XXXXXXXX
SENSOR OBSERVATIONS: YEAR: XXXX
                     EPOCH: DAY XXX  HOUR XX  MINUTE XX  SECOND XXXXXXXX
                     ASCENSION: XXXXXXXX
                     ECCENTRICITY: XXXXXXXX
                     INCLINATION: XXXXXXXX
                     ARGUMENT OF PERIGEE: XXXXXXXX
                     MEAN ANOMALY: XXXXXXXX
                     MEAN MOTION: XXXXXXXX
                     N DOT / 2 : XXXXXXXX
                     N DOUBLE DOT / 6: XXXXXXXX
```

FIGURE A-55
GROUND BASED SENSOR INPUTS MODIFY AND ADD RECORD INPUT SCREEN

```
SENSOR IDENTIFICATION NUMBER xxxxxxxx  SENSOR NAME  xxxxxxxx  PAGE 1:2
LAUNCH IDENTIFICATION NUMBER xxxxxxxx
LAUNCH DATE: MONTH xx DAY xx YEAR xxxx LAUNCH TIME:HOUR xx MIN xx SEC xxxxxxxx
LAUNCH LOCATION: LATITUDE xxxxx LONGITUDE xxxxx  ALTITUDE xxxxx
SENSOR OBSERVATIONS:  TIME-(HHMMSS)   INTENSITY   AZIMUTH-(RAD)  ELEVATION-(RAD)
                (1):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
                (2):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
                (3):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
                (4):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
                (5):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
                (6):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
                (7):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
                (8):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
                (9):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (10):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (11):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (12):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (13):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (14):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (15):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (16):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (17):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (18):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (19):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
                                                  PAGE 2:2
SENSOR OBSERVATIONS:  TIME-(HHMMSS)   INTENSITY   AZIMUTH-(RAD)  ELEVATION-(RAD)
               (20):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (21):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (22):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (23):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (24):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (25):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (26):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
               (27):  XXXXXXXX        XXXXXXXX    XXXXXXXX         XXXXXXXX
```

Figure A-56 IR Inputs Modify and Add Record Input Screen

```
SENSOR IDENTIFICATION NUMBER xxxxxxxx SENSOR NAME  xxxxxxxx PAGE  1:2
LAUNCH IDENTIFICATION NUMBER xxxxxxxx
LAUNCH DATE: MONTH xx DAY xx YEAR xxxx LAUNCH TIME:HOUR xx MIN xx  SEC  xxxxxxxx
LAUNCH LOCATION: LATITUDE xxxxx  LONGITUDE  xxxxx  ALTITUDE xxxxx
SENSOR OBSERVATIONS:
              HH MM SS                        SECONDS
1)      TIME                INTERVAL LENGTH: xxxxxxxx

X(0)=  xxxxxxxx    X(1)=  xxxxxxxx    X(2)=  xxxxxxxx    X(3)= xxxxxxxx
Y(0)=  xxxxxxxx    Y(1)=  xxxxxxxx    Y(2)=  xxxxxxxx    Y(3)= xxxxxxxx
Z(0)=  xxxxxxxx    Z(1)=  xxxxxxxx    Z(2)=  xxxxxxxx    Z(3)= xxxxxxxx

2)      TIME                INTERVAL LENGTH: xxxxxxxx

X(0)=  xxxxxxxx    X(1)=  xxxxxxxx    X(2)=  xxxxxxxx    X(3)= xxxxxxxx
Y(0)=  xxxxxxxx    Y(1)=  xxxxxxxx    Y(2)=  xxxxxxxx    Y(3)= xxxxxxxx
Z(0)=  xxxxxxxx    Z(1)=  xxxxxxxx    Z(2)=  xxxxxxxx    Z(3)= xxxxxxxx

3)      TIME                INTERVAL LENGTH: xxxxxxxx

X(0)=  xxxxxxxx    X(1)=  xxxxxxxx    X(2)=  xxxxxxxx    X(3)= xxxxxxxx
Y(0)=  xxxxxxxx    Y(1)=  xxxxxxxx    Y(2)=  xxxxxxxx    Y(3)= xxxxxxxx
Z(0)=  xxxxxxxx    Z(1)=  xxxxxxxx    Z(2)=  xxxxxxxx    Z(3)= xxxxxxxx
                                                             PAGE 2:2
              HH MM SS                        SECONDS
4)      TIME                INTERVAL LENGTH: xxxxxxxx

X(0)=  xxxxxxxx    X(1)=  xxxxxxxx    X(2)=  xxxxxxxx    X(3)= xxxxxxxx
Y(0)=  xxxxxxxx    Y(1)=  xxxxxxxx    Y(2)=  xxxxxxxx    Y(3)= xxxxxxxx
Z(0)=  xxxxxxxx    Z(1)=  xxxxxxxx    Z(2)=  xxxxxxxx    Z(3)= xxxxxxxx

5)      TIME                INTERVAL LENGTH: xxxxxxxx

X(0)=  xxxxxxxx    X(1)=  xxxxxxxx    X(2)=  xxxxxxxx    X(3)= xxxxxxxx
Y(0)=  xxxxxxxx    Y(1)=  xxxxxxxx    Y(2)=  xxxxxxxx    Y(3)= xxxxxxxx
Z(0)=  xxxxxxxx    Z(1)=  xxxxxxxx    Z(2)=  xxxxxxxx    Z(3)= xxxxxxxx
```

Figure A-57 Polynomial Inputs Modify and Add Record Input Screen

Select Launch Identification Number

Pressing the SELECT LAUNCH ID soft key initiates setting the default launch identification number. The analyst is asked to enter the launch identification number by the screen depicted in Figure A-58 (the current launch identification number is presented as the default). At this point, the analyst can enter the identification number of the launch folder of an event which has already taken place or of a new launch event. If a new launch identification number is specified, a blank launch folder is created. Whether a new or old launch identification number is specified, the application then displays all the information stored in the folder for this launch event in the screen depicted by Figure A-48. The analyst may then update or modify any entries in the launch folder. The launch identification number becomes the system default current launch event identification number.

Select Payload Identification Number

Pressing the SELECT PAYLOAD ID soft key initiates this application. The analyst is asked to enter the payload identification number by the screen depicted in Figure A-59 (the current payload identification number is presented as the default). If the selected payload is not found after scanning all the launch folders, the analyst is informed of this by the displaying of the message "THERE ARE NO LAUNCH FOLDERS MATCHING THE CONDITIONS SPECIFIED". Otherwise, the application displays all the information stored in the launch folder for this launch event in the screen depicted by Figure A-48. The analyst may then update or modify any entries in the launch folder. The launch identification number becomes the system default current event identification number.

THE LAUNCH I.D. TO BE USED IS: XXXXXXXX

Figure A-58 Select Launch Identification Number Input Screen

THE PAYLOAD I.D. TO BE USED IS: XXXXXXXX

Figure A-59 Select Payload Identification Number Input Screen

A.3.4 Data Base Add Function

Pressing the ADD A NEW RECORD soft key initiates the add function. This allows the analyst to enter a new record to a data base whenever a new capability is developed. The analyst is presented with the screen depicted in Figure A-60, with the last data base reviewed indicated as the default data base to be added. After the data base is selectd, a screen of a blank record for that data base is presented to the analyst upon which he may enter the new information. The screen presented depends upon the data base being added to.

If the data base to be added is	The input record screen is Figure
"launch folder"	A-48
"launch vehicle"	A-49
"launch site"	A-50
"Blue radar"	A-51
"Blue spaceborne sensor"	A-52
"Soviet ESV status"	A-53
"tracking facilities"	A-54
"ground based sensor inputs"	A-55
"IR inputs"	A-56
"polynomial inputs"	A-57

If the analyst wishes to add a new "launch folder" data base record and to set this new launch event as the current launch event, the analyst should select the launch identification number as defined in Section A.3.3.2, under "Select Launch Identification."

```
WHICH DATA BASE IS TO BE ADDED
 1 - LAUNCH FOLDER
 2 - IR INPUTS
 3 - LAUNCH VEHICLES
 4 - BLUE SPACEBORNE SENSOR SYSTEMS
 5 - LAUNCH SITE FILE
 6 - TRACKING AND RECIEVING SUPPORT FACILITIES
 7 - BLUE GROUND BASED SENSOR SYSTEMS
 8 - SOVIET ESV STATUS FILE
 9 - GROUND BASED SENSOR INPUTS
10 - POLYNOMIAL FILE

ENTER NUMBER OF THE DATA BASE TO BE ADDED     XX
```

Figure A-60 Add Input Screen

A.3.5 Data Base Delete Function

Pressing the DELETE AN EXISTING RECORD soft key initiates the delete function. When this application is selected, the analyst is presented with an initial transaction screen, which depends on the last data base reviewed.

If the last data base reviewed was	The input screen is Figure
"launch folder"	A-38
"launch vehicle"	A-39
"launch site"	A-40
"Blue radar"	A-41
"Blue spaceborne sensor"	A-42
"Soviet ESV status"	A-43
"tracking facilities"	A-44
"ground based sensor inputs"	A-45
"IR inputs"	A-46
"polynomial inputs"	A-47

At the top of the input screen, the analyst is informed what the last data base reviewed was. At the bottom of the screen, the application outputs the unique characteristics of the last record that was reviewed. This application is designed with the philosophy that the analyst will most likely want to delete the last record he reviewed.

If the analyst wishes to delete a record other than the one which was last reviewed, he merely has to change the unique characteristics shown at the bottom of the screen. If the analyst wishes to delete a record in some other data base, he has to modify the data base identifier at the top of the screen. If the data base identifier is modified, a new transaction screen is presented to the analyst according to the data base to be deleted.

If the data base to be deleted is	The input screen is Figure
"launch folder"	A-38
"launch vehicle"	A-39
"launch site"	A-40
"Blue radar"	A-41
"Blue spaceborne sensor"	A-42
"Soviet ESV status"	A-43
"tracking facilities"	A-44
"ground based sensor inputs"	A-45
"IR inputs"	A-46
"polynomial inputs"	A-47

The data base identifier at the top of the screen indicates which data base is to be deleted. However, since the record to be deleted is not the last record to be reviewed, no characteristics are presented to the analyst as defaults at the bottom of the screen. The analyst must enter the unique characteristics of the record to be deleted. After the unique characteristics have been entered, the record described is deleted from the data base described.

A.4 MAP APPLICATIONS

SABERS attempts to provide many graphical tools to aid the analyst in visualizing the geometry of space and missile events. These aids include a flexible map drawing capability and overlay capabilities such as drawing a satellite ground trace, a facility's location and coverage, and a reconnaissance satellite's coverage. An overlay graphic application does not erase the current graphic display before output, but adds the output to the current display.

A.4.1 Map Drawing Applications

The map drawing applications provide the analyst with the capability to draw the map with as much or as little detail as he wishes. In addition, the analyst may add political boundaries and/or a map grid at a later time without disturbing the rest of the map. The analyst also may redraw the map according to the parameters specified the last time he drew the map.

A.4.1.1 Display a World Map Application

Pressing the DISPLAY A WORLD MAP soft key initiates the map drawing application. The analyst is presented with the transaction screen depicted in Figure A-61. The analyst defines the map characteristics by entering the projection type from the list of Miller, Mercator, equirectangular, sinusoidal and orthographic. The analyst specifies the area of the world to be displayed by entering the latitude and longitude ranges. He defines the point on the earth above which the observer is situated at infinity by entering the center point latitude and longitude for the orthographic projection only. The analyst specifies the resolution of the map by entering the point interval to be plotted, and also indicates whether the political boundaries and/or map grid should be drawn.

After the analyst has specified the map parameters, the system stores them away in case the analyst may later wish to redraw the map (see Section A.4.1.4). Then the graphics display is erased (the map drawing application and the map redrawing application output the maps in the new frame graphics mode) and the map is then plotted. Examples of input and output combinations are presented in Figures A-62 to A-71. Figure A-62 represents the default applications offered by this application to the analyst.

```
PROJECTION TYPE:   X
      1 = MILLER
      2 = MERCATOR
      3 = EQUIRECTANGULAR
      4 = SINUSOIDAL
      5 = ORTHOGRAPHIC
LATITUDE RANGE:    XXXXX  TO   XXXXX
LONGITUDE RANGE:   XXXXX  TO   XXXXX
FOR ORTHOGRAPHIC PROJECTION ONLY:
   CENTER POINT- LAT:  XXXXX  LONG:  XXXXX
TRUE SCALE LATITUDE:   XXXXX
PLOT EVERY XX TH POINT
PLOT POLITICAL BOUNDARIES (Y OR N): X
MAP GRID (Y OR N): X
```

Figure A-61 Map Input Screen

```
PROJECTION TYPE: 1
     1 = MILLER
     2 = MERCATOR
     3 = EQUIRECTANGULAR
     4 = SINUSOIDAL
     5 = ORTHOGRAPHIC
LATITUDE RANGE:  -90.000 TO  90.000
LONGITUDE RANGE: -180.00 TO  180.00
FOR ORTHOGRAPHIC PROJECTION ONLY:
   CENTER POINT- LAT:  90.00 LONG: -105.00
TRUE SCALE LATITUDE:  0.0000
PLOT EVERY 10TH POINT
PLOT POLITICAL BOUNDARIES (Y OR N): N
MAP GRID (Y OR N): N
```

Figure A-62 Default Map Options Input Screen (Miller Projection)

Figure A-63 Default Map Options Output (Miller Projection)

```
PROJECTION TYPE: 2
     1 = MILLER
     2 = MERCATOR
     3 = EQUIRECTANGULAR
     4 = SINUSOIDAL
     5 = ORTHOGRAPHIC
LATITUDE RANGE:  -90.000 TO  90.000
LONGITUDE RANGE: -180.00 TO  180.00
FOR ORTHOGRAPHIC PROJECTION ONLY:
   CENTER POINT- LAT:  90.00 LONG: -105.00
TRUE SCALE LATITUDE:  0.0000
PLOT EVERY 1 TH POINT
PLOT POLITICAL BOUNDARIES (Y OR N): Y
MAP GRID (Y OR N): N
```

Figure A-64 Example of Fine Resolution Mercator Projection with Political Boundaries Input Screen

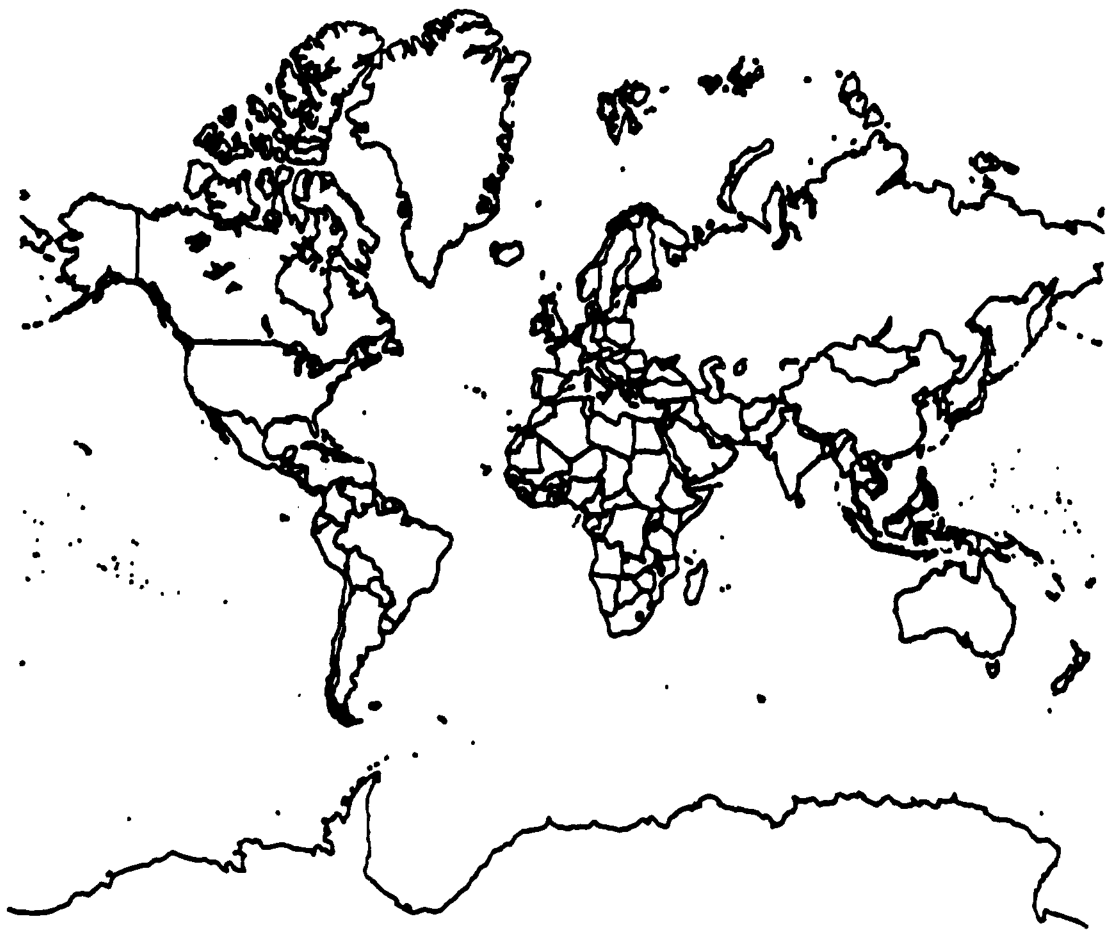


Figure A-65 Example of Fine Resolution Mercator Projection with Political Boundaries Output

```
PROJECTION TYPE: 3
     1 = MILLER
     2 = MERCATOR
     3 = EQUIRECTANGULAR
     4 = SINUSOIDAL
     5 = ORTHOGRAPHIC
LATITUDE RANGE:  -90.000 TO  90.000
LONGITUDE RANGE: -180.00 TO  180.00
FOR ORTHOGRAPHIC PROJECTION ONLY:
   CENTER POINT- LAT:  90.00 LONG: -105.00
TRUE SCALE LATITUDE:  0.0000
PLOT EVERY 30TH POINT
PLOT POLITICAL BOUNDARIES (Y OR N): N
MAP GRID (Y OR N): N
```

Figure A-66 Example of Coarse Resolution Equirectangular Projection Input Screen

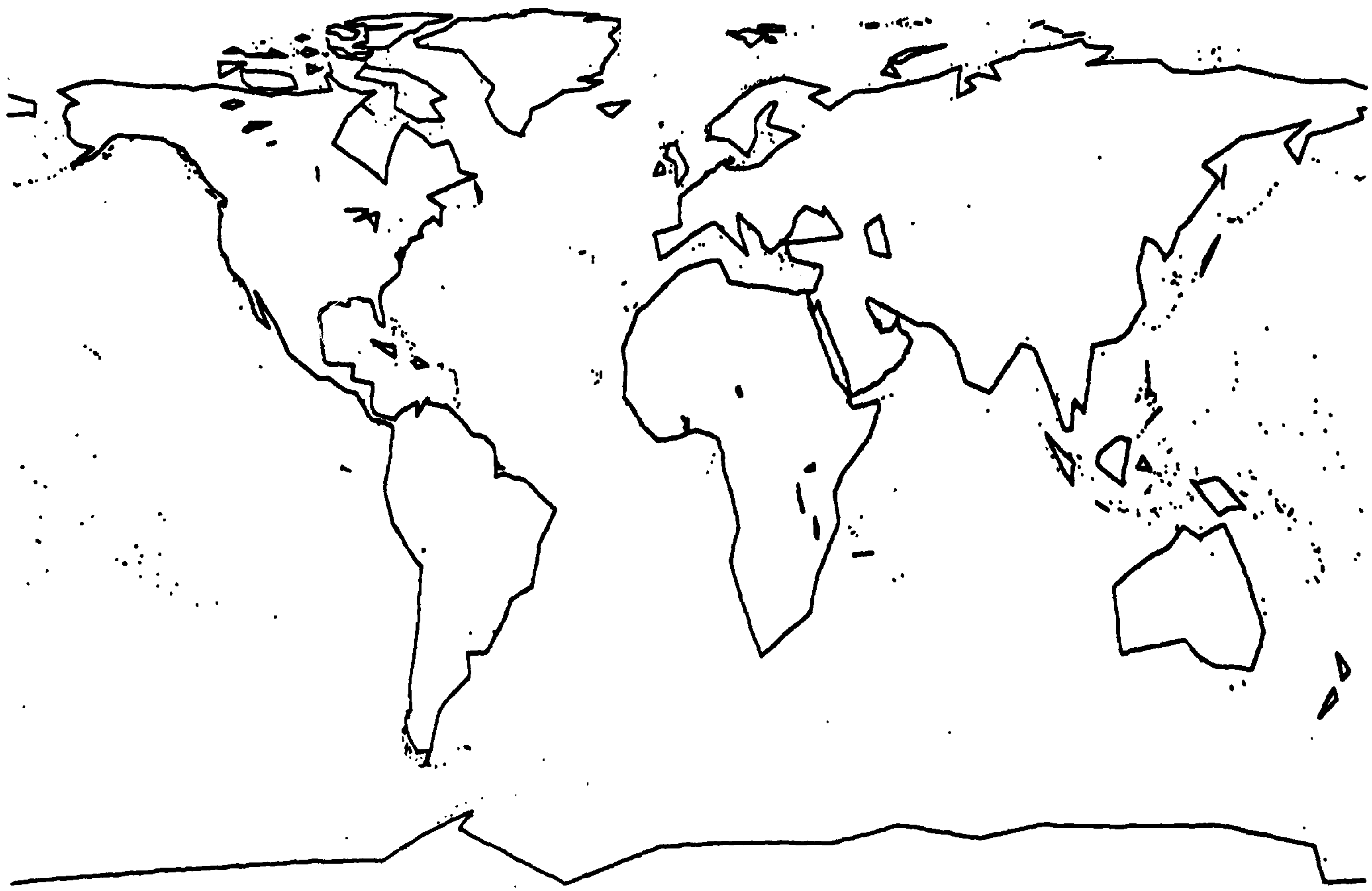


Figure A-67 Example of Coarse Resolution Equirectangular Projection Output

```
PROJECTION TYPE: 4
     1 = MILLER
     2 = MERCATOR
     3 = EQUIRECTANGULAR
     4 = SINUSOIDAL
     5 = ORTHOGRAPHIC
LATITUDE RANGE:  -90.000 TO  90.000
LONGITUDE RANGE: -180.00 TO  180.00
FOR ORTHOGRAPHIC PROJECTION ONLY:
   CENTER POINT- LAT:  90.00 LONG: -105.00
TRUE SCALE LATITUDE:  0.0000
PLOT EVERY 10TH POINT
PLOT POLITICAL BOUNDARIES (Y OR N): Y
MAP GRID (Y OR N): Y
```

Figure A-68 Example of Sinusoidal Projection with Political Boundaries and Grid Input Screen

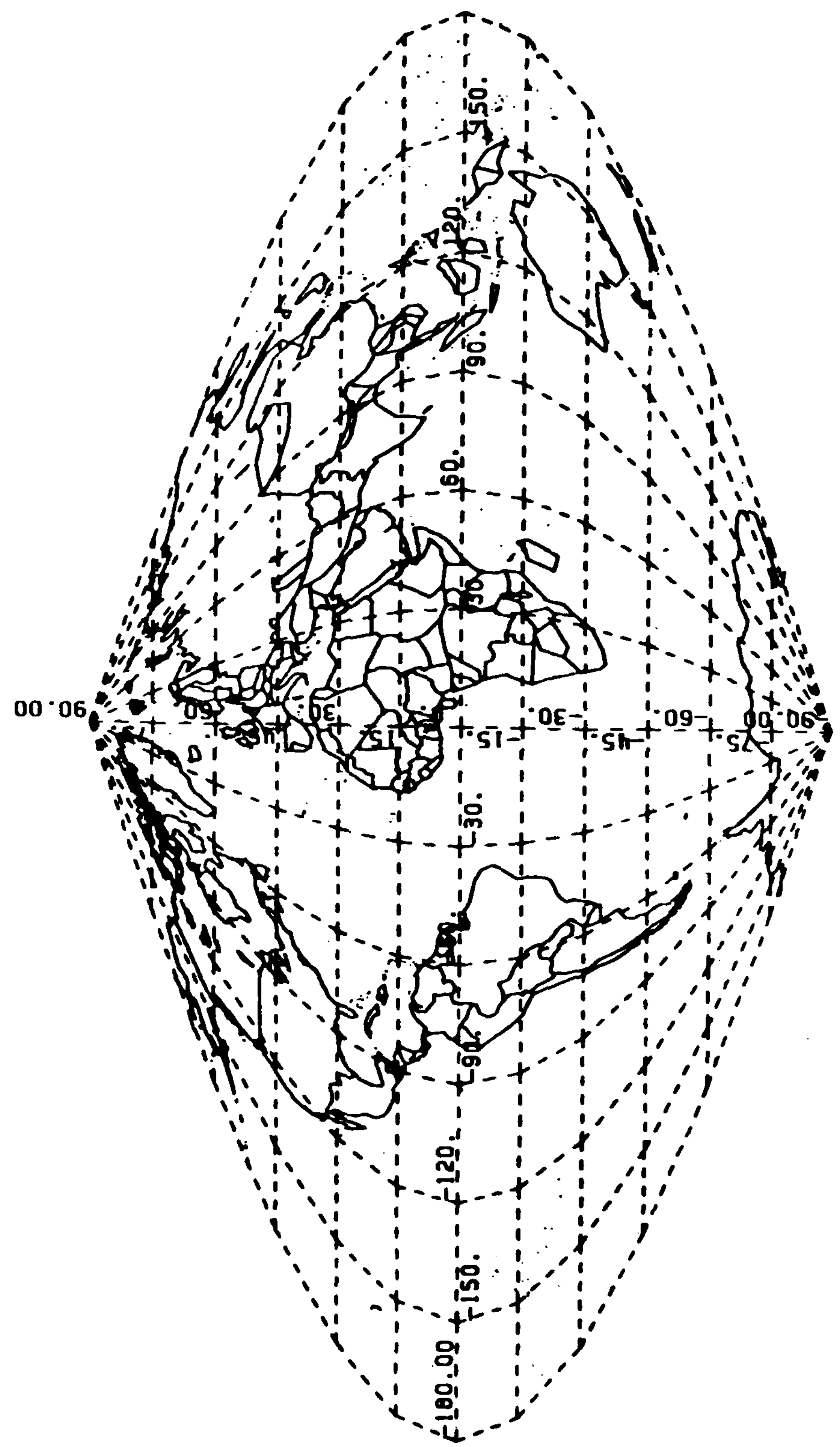


Figure A-69 Example of Sinusoidal Projection with Political Boundaries and Grid Output

```
PROJECTION TYPE: 5
     1 = MILLER
     2 = MERCATOR
     3 = EQUIRECTANGULAR
     4 = SINUSOIDAL
     5 = ORTHOGRAPHIC
LATITUDE RANGE:  -90.000 TO  90.000
LONGITUDE RANGE: -180.00 TO  180.00
FOR ORTHOGRAPHIC PROJECTION ONLY:
   CENTER POINT- LAT:  90.00 LONG: -105.00
TRUE SCALE LATITUDE:  0.0000
PLOT EVERY 10TH POINT
PLOT POLITICAL BOUNDARIES (Y OR N): Y
MAP GRID (Y OR N): N
```

Figure A-70 Example of Orthographic Projection with Political Boundaries Input Screen

Figure A-71 Example of Orthographic Projection with Political Boundaries Output

A.4.1.2 Draw Political Boundaries Application

Pressing the DRAW POLITICAL BOUNDARIES soft key results in the political boundaries being added to the current map display. There is no additional interaction with the analyst.

A.4.1.3 Draw Map Grid Application

Pressing the DRAW MAP GRID soft key results in the map grid being added to the current map display. There is no additional interaction with the analyst.

A.4.1.4 Redraw Map Only Application

Pressing the REDRAW MAP ONLY soft key results in the current graphic display being erased, and the map only being drawn according to the parameters input by the analyst when he last exercised the DISPLAY A WORLD map application. There is no additional interaction with the analyst. The analyst may add political boundaries and/or a map grid separately using the DRAW POLITICAL BOUNDARIES and/or DRAW MAP GRID applications.

A.4.2 Map Overlay Applications

The map overlay applications process the analyst inputs and generate a graphic output which is added to the current world map display. Each application has been designed to provide a pictorial representation of the geometry of occurrences of interest to the space and missile intelligence analyst.

The map overlay applications may be run in any order. Each output of a map overlay application is added to the plot already displayed. Note that in the examples presented for these applications, the world map upon which the application output is displayed is not drawn as a result of exercising the map overlay application. Rather, the world map is the result of running the DISPLAY A WORLD MAP application (see Section A.4.1.1)

A.4.2.1 Overlay Current Launch Point Application

The analyst may desire to graphically visualize the location of the current launch point. Pressing the OVERLAY CURRENT LAUNCH POINT initiates this application by extracting the launch point from the current launch folder and presenting the screen depicted in Figure A-72. The analyst may enter a different launch identification number, allowing the application to retrieve the launch point from the corresponding launch folder, or may erase the launch identification number, and enter any latitude and longitude of interest. As shown in the example output of Figure A-73, the point is designated by a marker and the launch identification number (if any).

A.4.2.2 Overlay Launch Sites Application

The analyst may desire to graphically visualize the locations of all the launch sites contained in the "launch site" data base. In particular, he may wish to see if any known launch sites overlap the current launch point plotted by the current launch point application (Section A.4.2.1). Pressing the

LAUNCH IDENTIFICATION NUMBER: XXXXXXXX

LATITUDE: XXXXXXXX LONGITUDE: XXXXXXXX

Figure A-72 Current Launch Point Input Screen

Figure A-73 Example of Current Launch Point Output

OVERLAY LAUNCH SITES soft key initiates this application. The application plots each launch site, along with the launch site name, on the current map display. There is no additional interaction with the analyst. An example output is presented in Figure A-74.

A.4.2.3 Overlay Red Support Facilities Application

The analyst may desire to graphically visualize the locations of all the Red tracking and support facilities contained in the "tracking facilities" data base. Pressing the OVERLAY RED SUPPORT FACILITIES soft key initiates this application. The application plots each Red tracking and support facility, along with the facility name, on the current map display. There is no additional interaction with the analyst. An example output is presented in Figure A-75.

A.4.2.4 Overlay Blue Radar Coverage Application

The analyst may desire to graphically visualize the locations, range, minimum azimuth and maximum azimuth of each Blue ground based sensor in the "Blue radar" data base. Pressing the OVERLAY BLUE RADAR COVERAGE soft key initiates this application. The application outputs the radar site name, and plots the projection of the radar range along a fan from the minimum azimuth to the maximum azimuth on the current world map. There is no additional interaction with the analyst. Example outputs are presented in Figures A-76 and A-77. Figure A-77 illustrates the problems caused by over-the-pole coverage on flat projections.

A.4.2.5 Overlay Satellite Ground Trace Application

The analyst may desire to graphically visualize the ground trace of a space object in its orbit or of a missile in its trajectory. The ground trace is defined to be the location on the earth such that a line perpendicular to the earth passes through the object. Pressing the OVERLAY GROUND TRACE soft

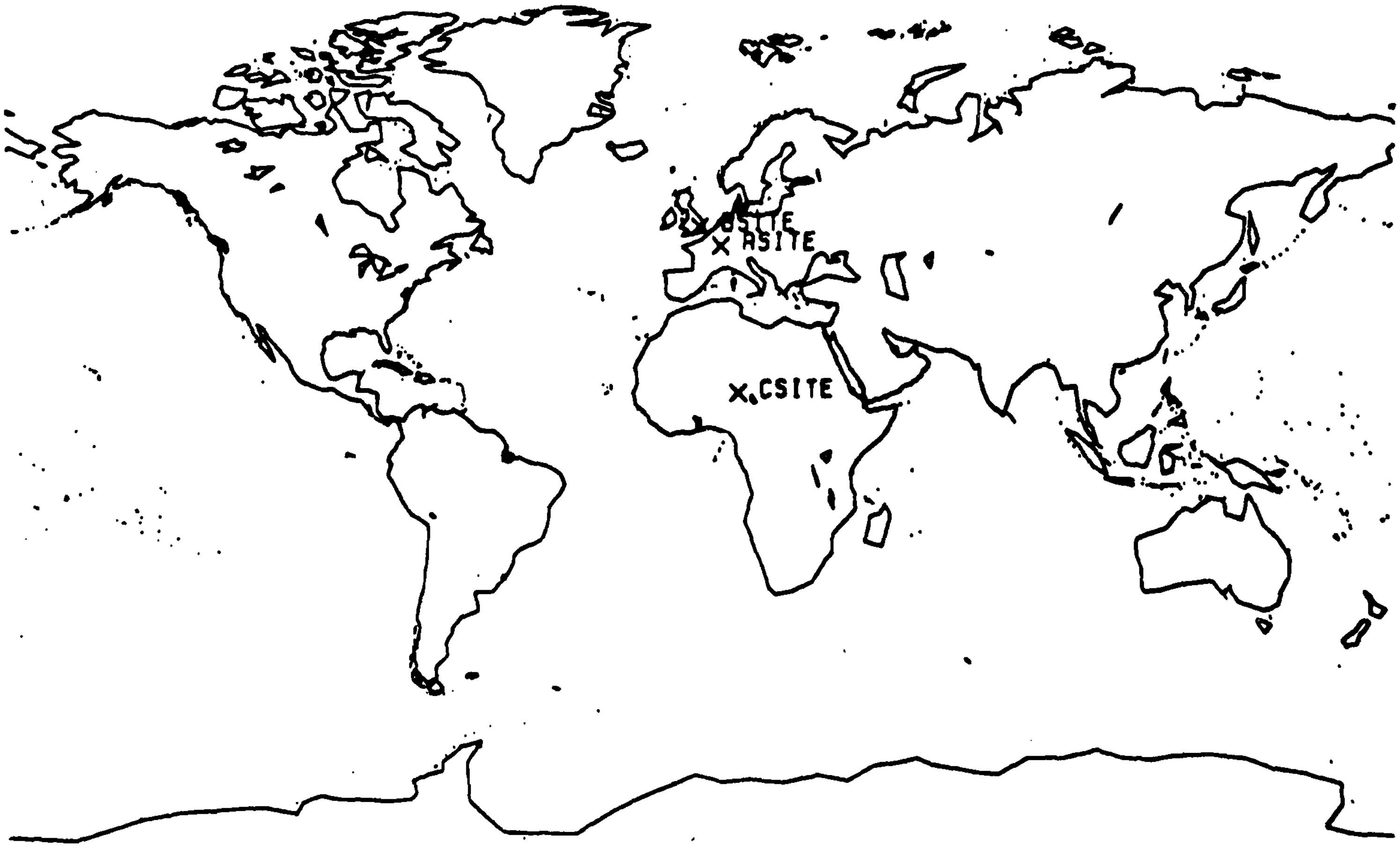
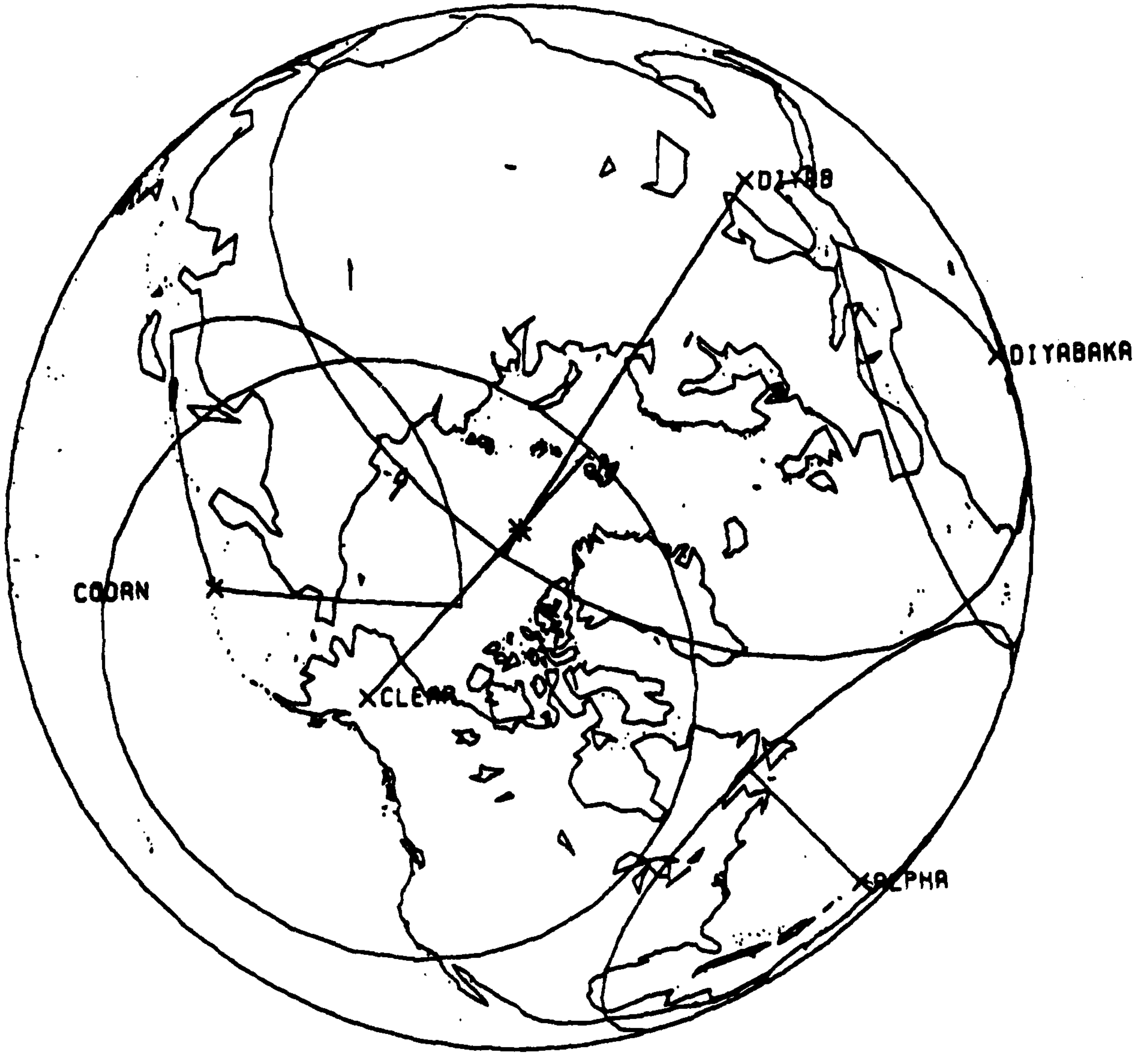


Figure A-74 Example of Launch Sites Display Output

Figure A-75 Example of Tracking and Receiving Support Facilities Display Output

Figure A-76 Example of Blue Ground Based Sensor Systems Display on Orthographic Projection



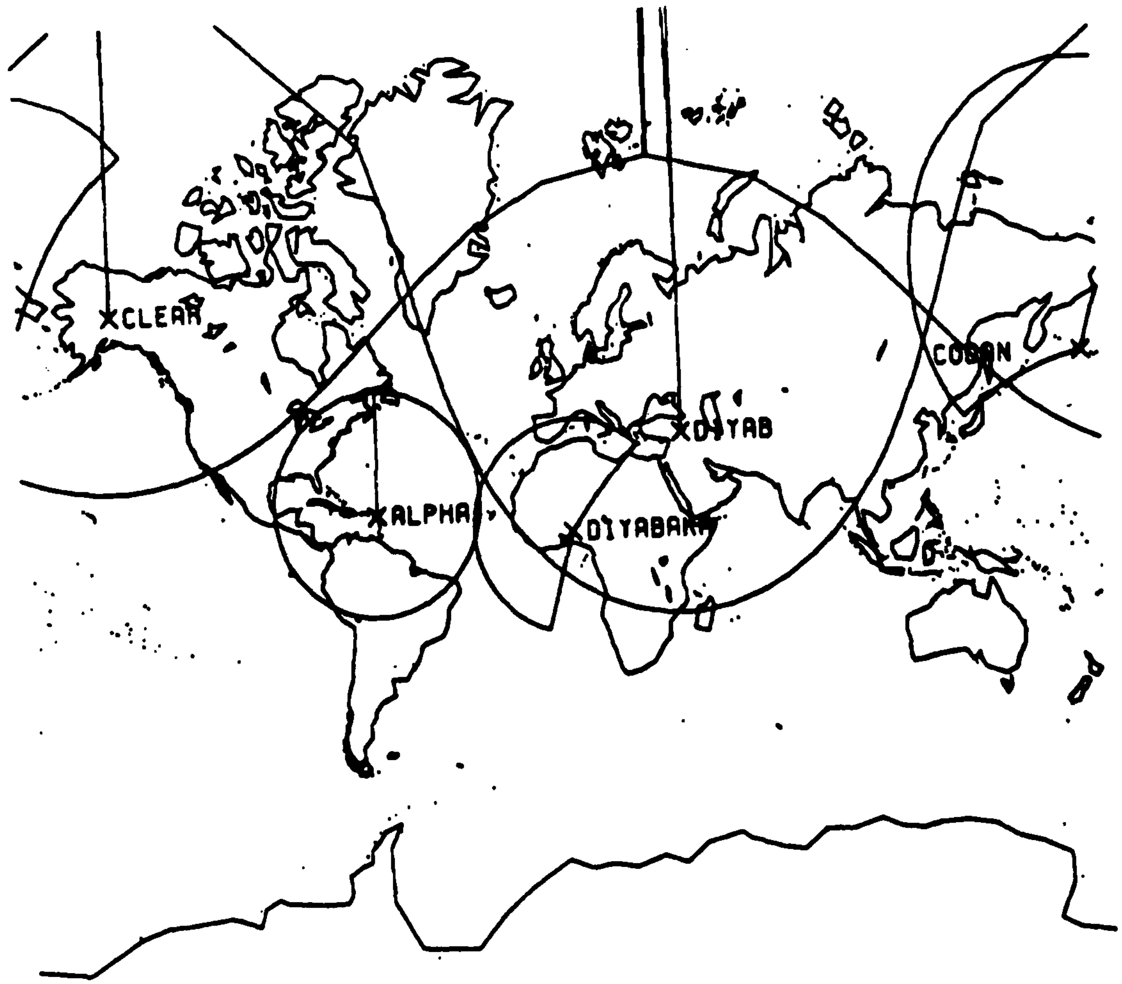


Figure A-77 Example of Blue Ground Based Sensor Systems Display on Mercator Projection

key initiates this application. The application determines the current time from the computer's clock, and sets this time as the start time. The end time is tentatively set as 24 hours later. The application then retrieves the launched satellite identification number from the current launch folder, and extracts the orbital element set for this satellite from the "ground based sensor inputs" data base. The orbital element set retrieved is the set whose epoch is the closest to the start time without being later than the start time. Finally, the application sets the end time to be that of one revolution expressed in three ways (number of revolutions, end time, and delta time), and presents the screen depicted in Figure A-78 to the analyst.

The analyst may change the satellite identification number. In this case, the application alters all the entries in the subsequent fields as it duplicates the actions described above for the new identification number. The analyst may also change the start time. In this case, if the "SATELLITE IDENTIFICATION NUMBER:" field is not blank, the application alters all the other fields as it searches for the new orbital element set satisfying the time constraint. Changing any other field results in the application using the new value in the ensuing calculations.

The application exits if the start time is greater than the end time. Otherwise, the application adds the requested ground trace to the current display. The satellite identification number (if any) is also output on the display. In addition, the time for one revolution is output at the bottom of the transaction screen. An example of the graphic output of this application is presented in Figure A-79.

A.4.2.6 Overlay Time Marks On Satellite Ground Trace Application

The analyst may desire to graphically visualize the time spent by a space object in each part of its orbit. This may be accomplished by the superposition of equal time-spaced tic marks on the ground trace of a space object in its orbit. Pressing the OVERLAY TIME MARKS ON GROUND TRACE soft key

```
TARGET ORBIT:
      SATELLITE IDENTIFICATION NUMBER:  XXXXXXXX
            EPOCH:   YEAR: XXXX   DAY: XXX  HOUR: XX   MIN: XX   SEC: XXXXXXXX
            RIGHT ASCENSION       : XXXXXXXX
            ECCENTRICITY          : XXXXXXXX
            INCLINATION           : XXXXXXXX
            ARGUMENT OF PERIGEE   : XXXXXXXX
            MEAN ANOMALY          : XXXXXXXX
            MEAN MOTION           : XXXXXXXX
            FIRST TIME DERIVATIVE
               OF MEAN MOTION     : XXXXXXXX
            SECOND TIME DERIVATIVE
               OF MEAN MOTION     : XXXXXXXX

PERIOD OF INTEREST:
      START TIME:   YEAR: XXXX   DAY: XXX   HOUR: XX   MIN: XX   SEC: XXXXXXXX
      END TIME (CHOOSE ONE):
                    NUMBER OF REVOLUTIONS: XXXXXXXX
                    YEAR: XXXX   DAY: XXX   HOUR: XX   MIN: XX   SEC: XXXXXXXX
                    DELTA TIME:  DAY: XXX   HOUR: XX   MIN: XX   SEC: XXXXXXXX
```

Figure A-78 Satellite Ground Trace, Time Marks and Radars Vs. Orbit Input Screen

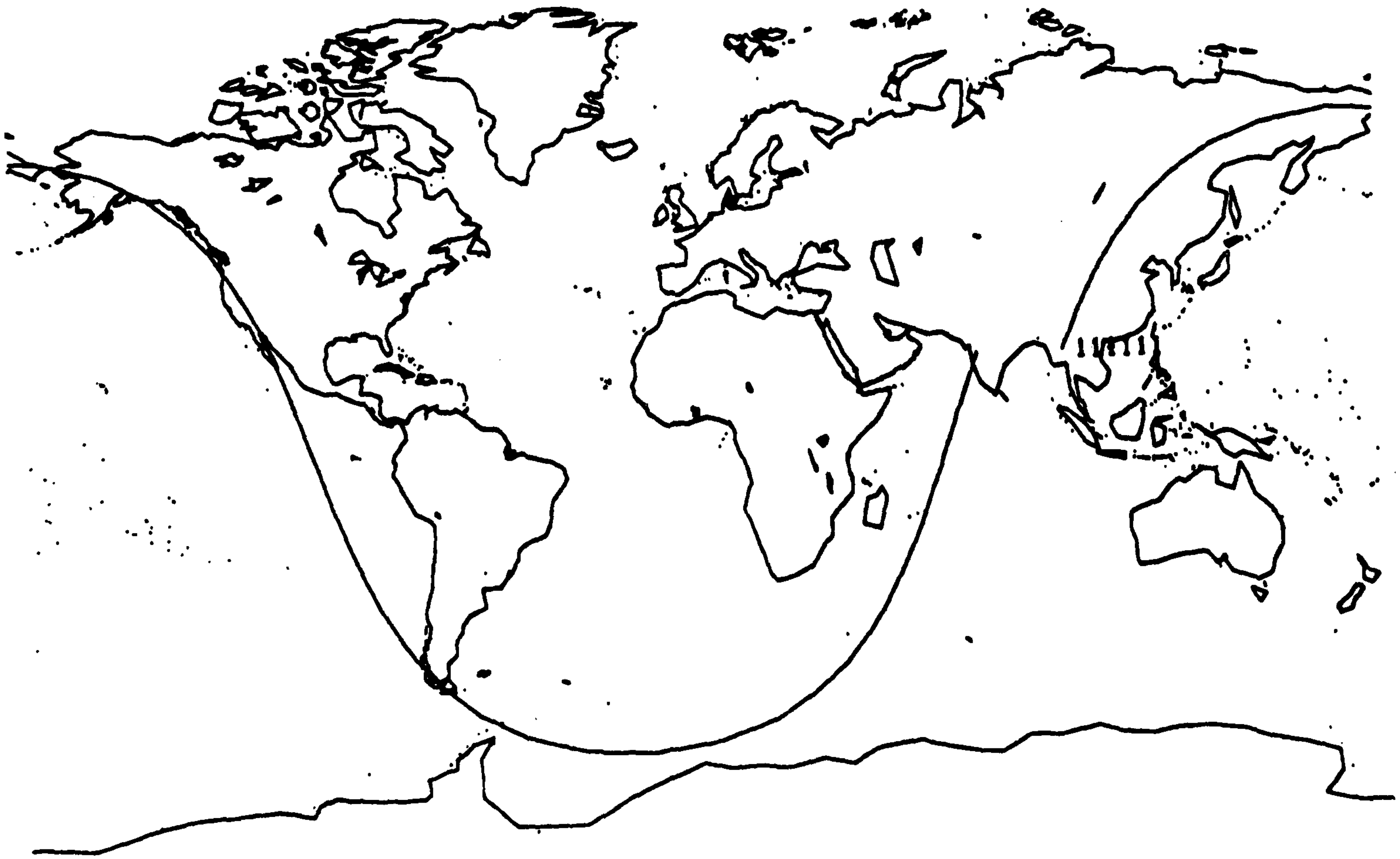


Figure A-79 Example of Satellite Ground Trace Output

initiates this application. The application determines the current time from the computer's clock, and sets this time as the start time. The end time is tentatively set as 24 hours later. The application then retrieves the launched satellite identification number from the current launch folder, and extracts the orbital element set for this satellite from the "ground based sensor inputs" data base. The orbital element set retrieved is the set whose epoch is the closest to the start time without being later than the start time. Finally, the application sets the end time to be that of one revolution expressed in three ways (number of revolutions, end time, and delta time), and presents the screen depicted in Figure A-78 to the analyst.

The analyst may change the satellite identification number. In this case, the application alters all the entries in the subsequent fields as it duplicates the actions described above for the new identification number. The analyst may also change the start time. In this case, if the "SATELLITE IDENTIFICATION NUMBER:" field is not blank, the application alters all the other fields as it searches for the new orbital element set satisfying the time constraint. Changing any other field results in the application using the new value in the ensuing calculations.

The application exits if the start time is greater than the end time or if the orbit element set is that of a ballistic missile trajectory. Otherwise, the application outputs 101 tic marks on the orbit ground trace in order to divide the orbit into 100 equal time-spaced parts. In addition, the time between tic marks is output at the bottom of the transaction screen. An example of the graphic output of this application is presented in Figure A-80.

A.4.2.7 Overlay Radars vs. Orbit Application

The analyst may desire to graphically visualize the Blue ground based sensors which may observe a space object in its orbit. This may be accomplished by plotting each radar which may observe the object. Pressing the OVERLAY RADARS VS. ORBIT soft key initiates this application. The

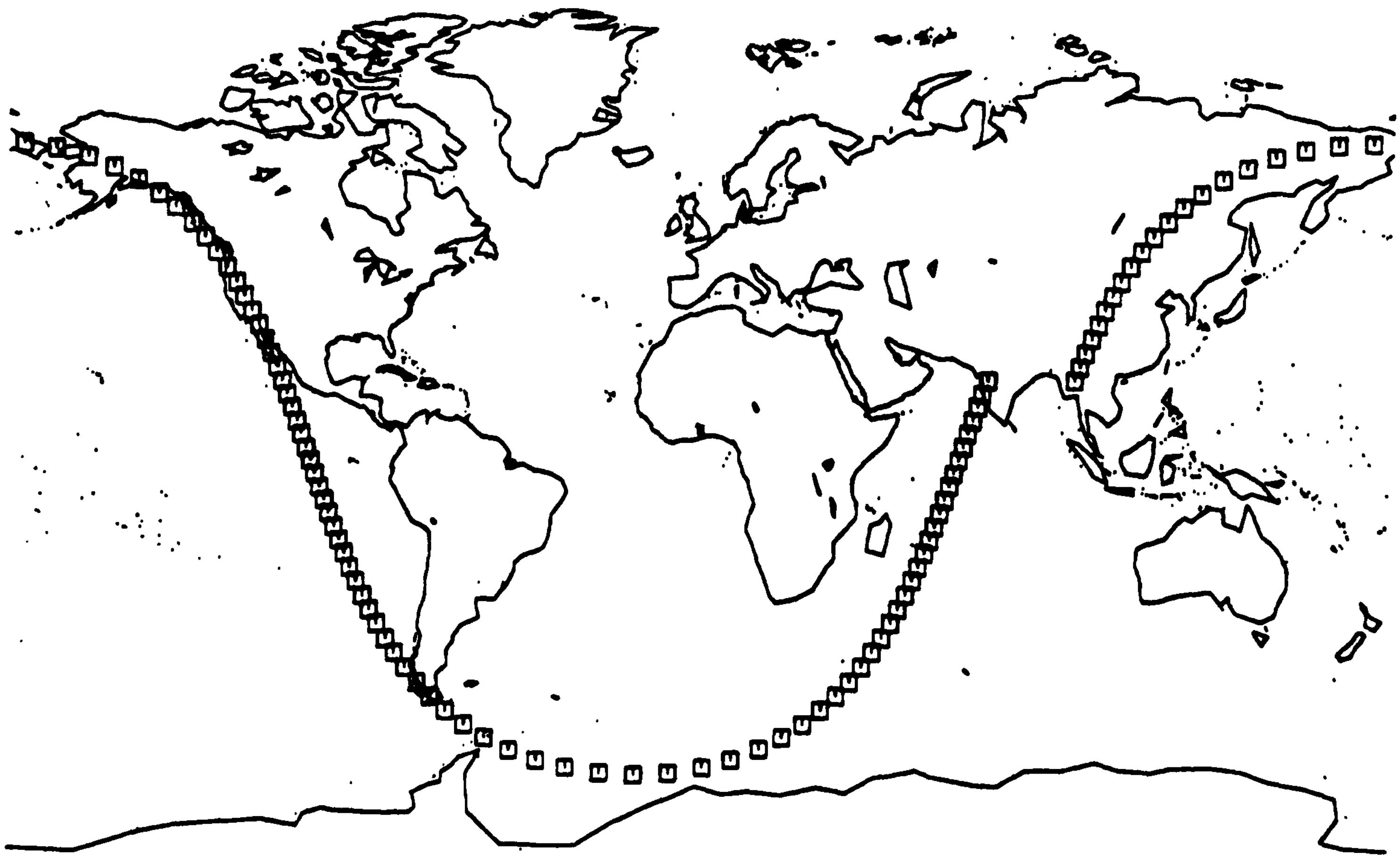


Figure A-80 Example of Time Marks Output

application determines the current time from the computer's clock, and sets this time as the start time. The end time is tentatively set as 24 hours later. The application then retrieves the launched satellite identification number from the current launch folder, and extracts the orbital element set for this satellite from the "ground based sensor inputs" data base. The orbital element set retrieved is the set whose epoch is the closest to the start time without being later than the start time. Finally, the application sets the end time to be that of one revolution expressed in three ways (number of revolutions, end time, and delta time), and presents the screen depicted in Figure A-78 to the analyst.

The analyst may change the satellite identification number. In this case, the application alters all the entries in the subsequent fields as it duplicates the actions described above for the new identification number. The analyst may also change the start time. In this case, if the "SATELLITE IDENTIFICATION NUMBER:" field is not blank, the application alters all the other fields as it searches for the new orbital element set satisfying the time constraint. Changing any other field results in the application using the new value in the ensuing calculations.

The application exits if the start time is greater than the end time or if the orbit element set is that of a ballistic missile trajectory. Otherwise, for each radar in the "Blue radar" data base, the application determines if the radar can observe the object in its orbit. If it can, the application outputs the radar site name, and plots the projection of the radar range along a fan from the smallest viewing azimuth to the largest viewing azimuth. An example of the graphic output of this application is presented in Figure A-81.

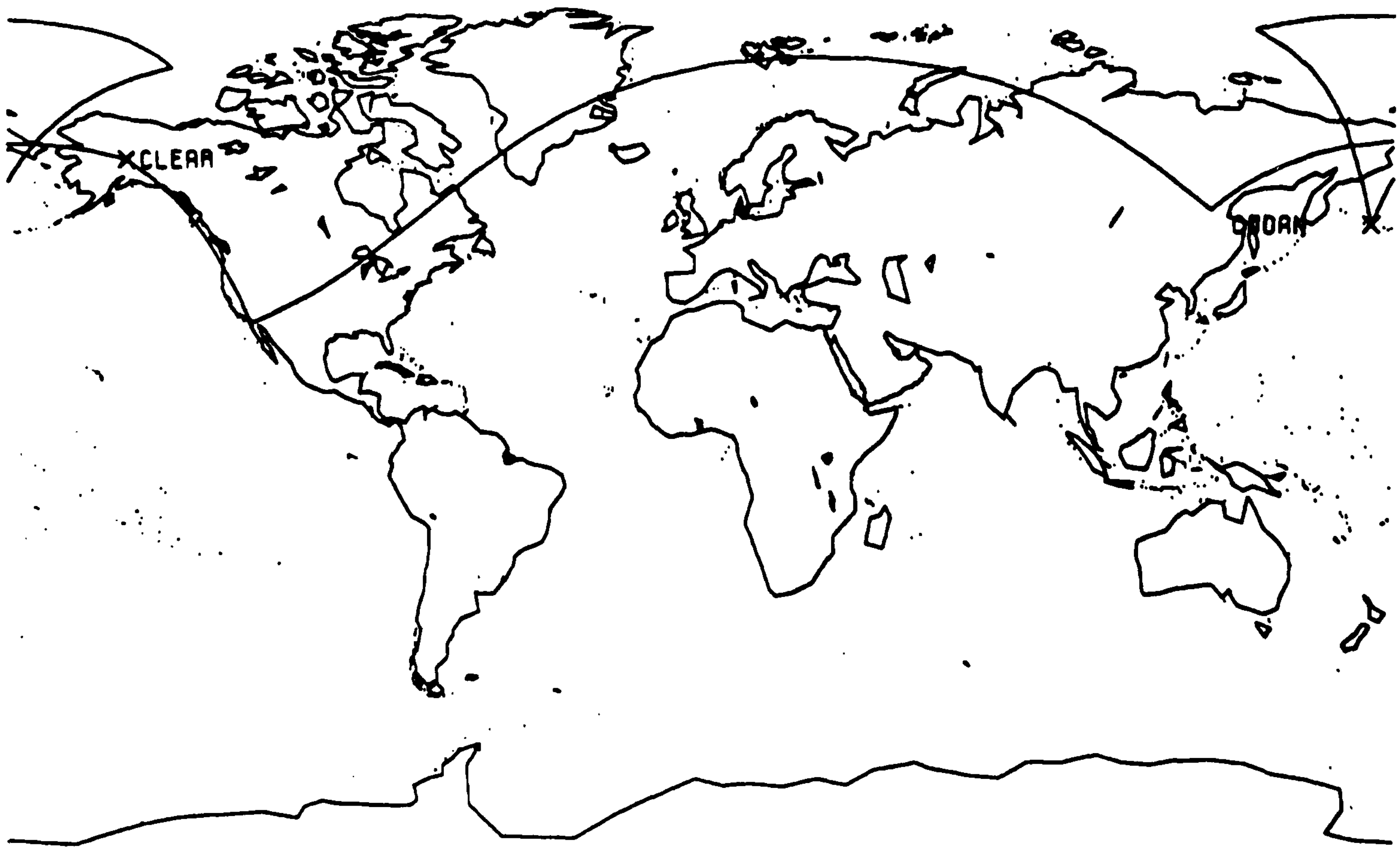


Figure A-81 Example of Radars Vs. Orbit Output

A.4.2.8 Overlay Satellite Photo Reconnaissance Application

The analyst may desire to graphically visualize the area on the earth's surface observed by a camera mounted on a photo reconnaissance satellite. This may be accomplished by the display of the intersection of the cone originating at the camera with the earth at equal time-spaced points of the satellite in its orbit. Pressing the OVERLAY SATELLITE RECONNAISSANCE soft key initiates this application. The application determines the current time from the computer's clock, and sets this time as the start time. The end time is tentatively set as 24 hours later. The application then retrieves the launched satellite identification number from the current launch folder, and extracts the orbital element set for this satellite from the "ground based sensor inputs" data base. The orbital element set retrieved is the set whose epoch is the closest to the start time without being later than the start time. The application sets the end time to be that of one revolution expressed in three ways (number of revolutions, end time, and delta time), and presents the screen depicted in Figure A-82 to the analyst. Finally, the application sets the default camera field of view to 90 degrees, and points the camera straight down.

The analyst may change the satellite identification number. In this case, the application alters all the entries in the subsequent fields as it duplicates the actions described above for the new identification number. The analyst may also change the start time. In this case, if the "SATELLITE IDENTIFICATION NUMBER:" field is not blank, the application alters all the other fields as it searches for the new orbital element set satisfying the time constraint. Changing any other field results in the application using the new value in the ensuing calculations.

The application exits if the start time is greater than the end time or if the orbit element set is that of a ballistic missile trajectory. Otherwise, the application adds the ground trace tic mark and the intersection of the cone with the earth (if any) for each equal time interval. An example

```
TARGET ORBIT:
      SATELLITE IDENTIFICATION NUMBER:  XXXXXXXX
            EPOCH:   YEAR: XXXX   DAY: XXX  HOUR: XX   MIN:  XX   SEC: XXXXXXXX
            RIGHT ASCENSION          : XXXXXXXX
            ECCENTRICITY             : XXXXXXXX
            INCLINATION              : XXXXXXXX
            ARGUMENT OF PERIGEE      : XXXXXXXX
            MEAN ANOMALY             : XXXXXXXX
            MEAN MOTION              : XXXXXXXX
            FIRST TIME DERIVATIVE
               OF MEAN MOTION        : XXXXXXXX
            SECOND TIME DERIVATIVE
               OF MEAN MOTION        : XXXXXXXX

PERIOD OF INTEREST:
      START TIME:    YEAR: XXXX   DAY: XXX   HOUR: XX   MIN: XX   SEC: XXXXXXXX
      END TIME (CHOOSE ONE):
                     NUMBER OF REVOLUTIONS: XXXXXXXX
                     YEAR: XXXX   DAY: XXX   HOUR: XX   MIN: XX   SEC: XXXXXXXX
                     DELTA TIME: DAY: XXX    HOUR: XX   MIN: XX   SEC: XXXXXXXX

CAMERA ANGLES:
      FIELD OF VIEW: XXXXXXXX   AZIMUTH: XXXXXXXX   ELEVATION: XXXXXXXX
```

Figure A-82 Satellite Photo Reconnaissance Input Screen

of the graphic output of this application is presented in Figure A-83.

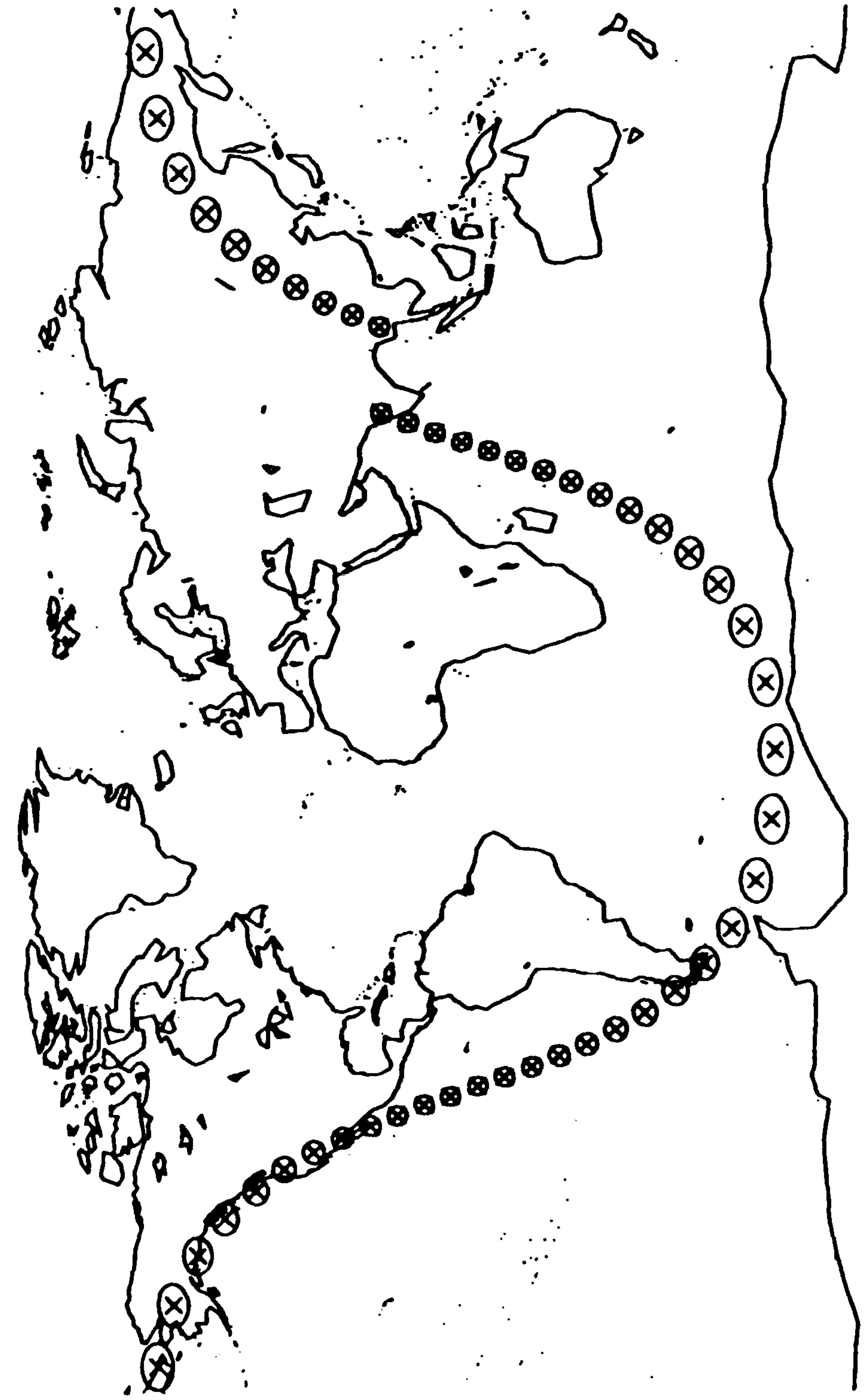


Figure A-83 Example of Satellite Photo Reconnaissance Output

A.5 LISTING GENERATING ANALYSIS APPLICATIONS

The analyst is frequently concerned with the possible time frames for space events. The SABERS applications which calculate time windows are the threat window generation capability, the radar vs. orbit capability and the photo reconnaissance coverage capability.

A.5.1 List Threat Windows Application

The analyst may desire to determine when a mission may be launched from a launch site with the intention of intercepting a target satellite in its orbit. This may be accomplished by calculating and listing the launch time window and the nominal launch time. Pressing the GENERATE THREAT WINDOWS soft key initiates this application. The application determines the current time from the computer's clock, and sets this time as the start time. The end time is set as 24 hours later. The application retrieves the launch site and the target satellite identification number from the current launch folder. The application extracts the launch site position from the "launch site" data base, and extracts the orbital element set for the satellite from the "ground based sensor inputs" data base. The orbital element set retrieved is the set whose epoch is the closest to the start time without being later than the start time. Finally, the application sets the end time to be one day later expressed in three ways (number of revolutions, end time, and delta time), and presents the screen depicted in Figure A-84 to the analyst.

The analyst may change the launch site name. In this case, the application alters the launch site position according to the location of the launch site in the "launch site" data base. The analyst may change the satellite identification number. In this case, the application alters all the entries in the subsequent fields as it duplicates the actions described above for the new identification number. The analyst may also change the start time. In this case, if the "SATELLITE IDENTIFICATION NUMBER:" field is not blank, the application alters all the other fields as it searches for the new

```
LAUNCH SITE NAME: XXXXXXXX
        LATITUDE: XXXXXXXX    LONGITUDE: XXXXXXXX    ALTITUDE: XXXXXXXX

TARGET ORBIT:
       SATELLITE IDENTIFICATION NUMBER: XXXXXXXX
             EPOCH:   YEAR: XXXX   DAY: XXX   HOUR: XX  MIN: XX  SEC: XXXXXXXX
             RIGHT ASCENSION       : XXXXXXXX
             ECCENTRICITY          : XXXXXXXX
             INCLINATION           : XXXXXXXX
             ARGUMENT OF PERIGEE   : XXXXXXXX
             MEAN ANOMALY          : XXXXXXXX
             MEAN MOTION           :
             FIRST TIME DERIVATIVE
                OF MEAN MOTION     : XXXXXXXX
             SECOND TIME DERIVATIVE
                OF MEAN MOTION     : XXXXXXXX

PERIOD OF INTEREST:
       START TIME:    YEAR: XXXX   DAY: XXX   HOUR: XX   MIN: XX   SEC: XXXXXXXX
       END TIME (CHOOSE ONE):
                      NUMBER OF REVOLUTIONS: XXXXXXXX
                      YEAR: XXXX   DAY: XXX   HOUR: XX   MIN: XX   SEC: XXXXXXXX
                      DELTA TIME:  DAY: XXX   HOUR: XX   MIN: XX   SEC: XXXXXXXX
```

Figure A-84 Threat Window Input Screen

orbital element set satisfying the time constraint. Changing any other field results in the application using the new value in the ensuing calculations.

The application exits if the start time is greater than the end time or if the orbit element set is that of a ballistic missile trajectory. Otherwise, the application generates the threat windows. If none exist, the message "NO THREAT WINDOWS EXIST" is output to the listing file. The message "RESULTS TABULATED IN FILE "THREAT.LIS"" is output at the bottom of the transaction screen. The listing may then be printed on the line printer or listed at the terminal by responding to the prompt with "THREAT.LIS" (see Section A.5.5). An example of the listing output of this application is presented in Figure A-85.

LAUNCH SITE	WINDOW OPEN TIME					NOMINAL LAUNCH TIME					WINDOW CLOSE TIME				
	YEAR	DAY	HR	MIN	SEC	YEAR	DAY	HR	MIN	SEC	YEAR	DAY	HR	MIN	SEC
ASITE	1980	260	9	11	39	1980	260	9	20	36	1980	260	9	25	27

Figure A-85 Example of Threat Window Listing

A.5.2 List Radars vs. Orbit Application

The analyst may desire to determine when each radar in the "Blue radar" data base may observe a space object in its orbit. This may be accomplished by calculating and listing the time windows of coverage. Pressing the LIST RADARS VS. ORBIT soft key initiates this application. The application determines the current time from the computer's clock, and sets this time as the start time. The end time is tentatively set as 24 hours later. The application then retrieves the launched satellite identification number from the current launch folder, and extracts the orbital element set for this satellite from the "ground based sensor inputs" data base. The orbital element set retrieved is the set whose epoch is the closest to the start time without being later than the start time. Finally, the application sets the end time to be that of one revolution expressed in three ways (number of revolutions, end time, and delta time), and presents the screen depicted in Figure A-78 to the analyst.

The analyst may change the satellite identification number. In this case, the application alters all the entries in the subsequent fields as it duplicates the actions described above for the new identification number. The analyst may also change the start time. In this case, if the "SATELLITE IDENTIFICATION NUMBER:" field is not blank, the application alters all the other fields as it searches for the new orbital element set satisfying the time constraint. Changing any other field results in the application using the new value in the ensuing calculations.

The application exits if the start time is greater than the end time or if the orbit element set is that of a ballistic missile trajectory. Otherwise, for each radar in the "Blue radar" data base, the application determines if the radar can observe the object in its orbit. If it can, the application outputs the radar site name, and lists the first and last time for each orbit that the radar can see the space object. If none of the radars can see the space object, the message "SATELLITE IS NOT UNDER RADAR COVERAGE" is

output to the listing file. The message "RESULTS TABULATED IN "ORBSEN.LIS" is output at the bottom of the transaction screen. The listing may then be printed on the line printer or listed at the terminal by responding to the prompt with "ORBSEN.LIS"" (see Section A.5.5). azimuth. An example of the listing output of this application is presented in Figure A-86.

SENSOR NAME	SENSOR SDC NUMBER	START YEAR	DAY	HR	MIN	SEC	STOP YEAR	DAY	HR	MIN	SEC
CLEAR	401	1980	259	13	2	11	1980	259	13	11	18
CODAN	402	1980	259	13	0	21	1980	259	13	3	60

Figure A-86 Example of Radars Vs. Orbit Listing

A.5.3 List Satellite Photo Reconnaissance Application

The analyst may desire to determine when each ground facility may be under satellite photo reconnaissance coverage. Presently, each radar in the "Blue radar" data base is used as the ground facility to be observed. This may be accomplished by calculating and listing the time windows of coverage. Pressing the LIST SATELLITE RECONNAISSANCE soft key initiates this application. The application determines the current time from the computer's clock, and sets this time as the start time. The end time is tentatively set as 24 hours later. The application then retrieves the launched satellite identification number from the current launch folder, and extracts the orbital element set for this satellite from the "ground based sensor inputs" data base. The orbital element set retrieved is the set whose epoch is the closest to the start time without being later than the start time. Finally, the application sets the end time to be that of one revolution expressed in three ways (number of revolutions, end time, and delta time), and presents the screen depicted in Figure A-82 to the analyst.

The analyst may change the satellite identification number. In this case, the application alters all the entries in the subsequent fields as it duplicates the actions described above for the new identification number. The analyst may also change the start time. In this case, if the "SATELLITE IDENTIFICATION NUMBER:" field is not blank, the application alters all the other fields as it searches for the new orbital element set satisfying the time constraint. Changing any other field results in the application using the new value in the ensuing calculations.

The application exits if the start time is greater than the end time or if the orbit element set is that of a ballistic missile trajectory. Otherwise, for each radar in the "Blue radar" data base, the application determines if the satellite camera can observe the ground facility. If it can, the application outputs the ground facility name, and lists the first and last time for each orbit that the satellite camera can observe the ground

facility. If none of the ground facilities can be seen by the camera, the message "FACILITIES ARE NOT UNDER SATELLITE PHOTO COVERAGE" is output to the listing file. The message "RESULTS TABULATED IN "COVERG.LIS"" is output at the bottom of the transaction screen. The listing may then be printed on the line printer or listed at the terminal by responding to the prompt with "COVERG.LIS" (see Section A.5.5). An example of the listing output of this application is presented in Figure A-87.

SENSOR NAME	SENSOR SDC	START	TIME	OF	VULNERABILITY				STOP		
	NUMBER	YEAR	DAY	HR	MIN	SEC	YEAR	DAY	HR	MIN	SEC
CLEAR	401	1980	259	13	5	50	1980	259	13	9	27

Figure A-87 Example of Satellite Photo Reconnaissance Listing

A.5.4 Listing and Overlay Applications

Two applications are represented in both the map overlay functions described in Section A.4.2 and the listing generating analysis functions of this section. These applications, the radars vs. orbit and satellite photo reconnaissance, have been combined to provide the further flexibility of both listing and graphic overlay output.

A.5.4.1 Radars vs. Orbit Application

The analyst may desire to determine when each radar in the "Blue radar" data base may observe a space object in its orbit, in addition to graphically visualizing the Blue ground based sensor. This may be accomplished by calculating and listing the time windows of coverage in conjunction with plotting each radar which may observe the object. Pressing the RADARS VS. ORBIT soft key initiates this application. The application determines the current time from the computer's clock, and sets this time as the start time. The end time is tentatively set as 24 hours later. The application then retrieves the launched satellite identification number from the current launch folder, and extracts the orbital element set for this satellite from the "ground based sensor inputs" data base. The orbital element set retrieved is the set whose epoch is the closest to the start time without being later than the start time. Finally, the application sets the end time to be that of one revolution expressed in three ways (number of revolutions, end time, and delta time), and presents the screen depicted in Figure A-78 to the analyst.

The analyst may change the satellite identification number. In this case, the application alters all the entries in the subsequent fields as it duplicates the actions described above for the new identification number. The analyst may also change the start time. In this case, if the "SATELLITE IDENTIFICATION NUMBER:" field is not blank, the application alters all the other fields as it searches for the new orbital element set satisfying the time constraint. Changing any other field results in the application using

the new value in the ensuing calculations.

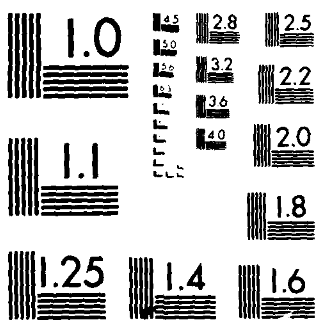
The application exits if the start time is greater than the end time or if the orbit element set is that of a ballistic missile trajectory. Otherwise, for each radar in the "Blue radar" data base, the application determines if the radar can observe the object in its orbit. If it can, the application outputs the radar site name to both the listing file and the graphic display, lists the first and last time for each orbit that the radar can see the space object to the output file, and plots the projection of the radar range along a fan from the smallest viewing azimuth to the largest viewing azimuth. If none of the radars can see the space object, the message "SATELLITE IS NOT UNDER RADAR COVERAGE" is output to the listing file. The message "RESULTS TABULATED IN "ORBSEN.LIS" is output at the bottom of the transaction screen. The listing may then be printed on the line printer or listed at the terminal by responding to the prompt with "ORBSEN.LIS" (see Section A.5.5). An example of the listing output of this application is presented in Figure A-86. An example of the graphic output of this application is presented in Figure A-81.

A.5.4.2 Satellite Photo Reconnaissance Application

The analyst may desire to determine when each ground facility may be under satellite photo reconnaissance coverage, in addition to graphically visualizing the area on the earth's surface observed by a camera mounted on a photo reconnaissance satellite. Presently, each radar in the "Blue radar" data base is used as the ground facility to be observed. This may be accomplished by calculating and listing the time windows of coverage in conjunction displaying the intersection of the cone originating at the camera with the earth at equal time-spaced points of the satellite in its orbit. Pressing the SATELLITE RECONNAISSANCE soft key initiates this application. The application determines the current time from the computer's clock, and sets this time as the start time. The end time is tentatively set as 24 hours later. The application then retrieves the launched satellite identification

AD-A110 272   PAR TECHNOLOGY CORP ROME NY   F/G 9/2
SABERS. STAND-ALONE ADIC BINARY EXPLOITATION RESOURCES SYSTEM. --ETC(U)
SEP 81 A J FRANKLIN, R L CALDWELL, S COLE   F30602-78-C-0078
UNCLASSIFIED   RADC-TR-81-250-VOL-2   NL

3 of 4
AD A
110272



MICROCOPY RESOLUTION TEST CHART

number from the current launch folder, and extracts the orbital element set for this satellite from the "ground based sensor inputs" data base. The orbital element set retrieved is the set whose epoch is the closest to the start time without being later than the start time. Finally, the application sets the end time to be that of one revolution expressed in three ways (number of revolutions, end time, and delta time), and presents the screen depicted in Figure A-82 to the analyst.

The analyst may change the satellite identification number. In this case, the application alters all the entries in the subsequent fields as it duplicates the actions described above for the new identification number. The analyst may also change the start time. In this case, if the "SATELLITE IDENTIFICATION NUMBER:" field is not blank, the application alters all the other fields as it searches for the new orbital element set satisfying the time constraint. Changing any other field results in the application using the new value in the ensuing calculations.

The application exits if the start time is greater than the end time or if the orbit element set is that of a ballistic missile trajectory. Otherwise, the application adds the ground trace tic mark and the intersection of the cone with the earth (if any) for each equal time interval. Then, for each radar in the "Blue radar" data base, the application determines if the satellite camera can observe the ground facility. If it can, the application outputs the ground facility name, and lists the first and last time for each orbit that the satellite camera can observe the ground facility. If none of the ground facilities can be seen by the camera, the message "FACILITIES ARE NOT UNDER SATELLITE PHOTO COVERAGE" is output to the listing file. The message "RESULTS TABULATED IN "COVERG.LIS"" is output at the bottom of the transaction screen. The listing may then be printed on the line printer or listed at the terminal by responding to the prompt with "COVERG.LIS" (see Section A.5.5). azimuth. An example of the listing output of this application is presented in Figure A-87. An example of the graphic output of this application is presented in Figure A-83.

A.5.5 Viewing The Listings

The analyst must interact with the operating system to view the output at either the terminal or the line printer. The commands are TYPE and PRINT, respectively. Two of the SABERS predefined soft keys output these commands when they are pressed. The operating system then prompts for the name of the file to be listed by typing on the terminal the message "$ FILE: ", requesting the analyst to input the required file name. Most of the SABERS applications which create listing files suitable for viewing either at the terminal or on the line printer output the name of the file created before completing execution.

A.5.5.1 Output To Line Printer

Pressing the HARDCOPY TO LINE PRINTER soft key directs the operating system to prepare to list a file on the line printer. The analyst enters the name of the file to be listed in response to the prompt message "$ FILE: ".

A.5.5.2 Output To Terminal

Pressing the VIEW AT TERMINAL soft key directs the operating system to prepare to list a file on the terminal. The analyst enters the name of the file to be listed in response to the prompt message "$ FILE: ".

A.6 GRAPHIC ANALYSIS APPLICATIONS

The applications discussed in this section are new frame graphic applications. This means that the graphics display is erased before the application begins its own graphic output. The purpose of these applications is to aid the analyst in solving the recognition problems of launch site and launch azimuth.

A.6.1 Zoom On Launch Point Application

Given a launch event and the reported launch position, the analyst may wish to determine which launch site, and which launch pad of the launch site, may have been the true launch position. Pressing the ZOOM ON LAUNCH SITE soft key initiates the application that will graphically aid the analyst in this determination. The application extracts the latitude, longitude and event type (space or missile) from the current "launch folder" data base record, sets the default ranges of latitude and longitude to one degree and the default error to zero kilometers. The analyst is then presented with this information in the transaction screen depicted in Figure A-88. The launch identification number may be changed, in which case the "launch folders" data base will be searched for the new launch position and event type. The analyst may display all the pads capable of launching space missions if the "EVENT TYPE:" field contains the word "space"; all the launch pads capable of launching missiles if the field contains "missile"; or all pads if the field is blank or contains the word "both". The range of latitude and longitude define the maximum difference between the reported launch position and the position of the closest launch pad. The error measure in kilometers defines the known inaccuracy of the reporting sensor.

An example of the graphic output is presented in Figure A-89. A line is drawn from the reported launch position to the closest launch pad. The circle centered at the reported launch position is drawn with its radius equal to the error measure. The launch site and launch pad indicated by this application

```
LAUNCH IDENTIFICATION NUMBER:
        LATITUDE:              DEG.
        LONGITUDE:             DEG.
        EVENT TYPE:
        RANGE OF LATITUDE:              DEG.    RANGE OF LONGITUDE:             DEG.
        ERROR:          KM.
```

Figure A-88 Zoom on Launch Point Input Screen

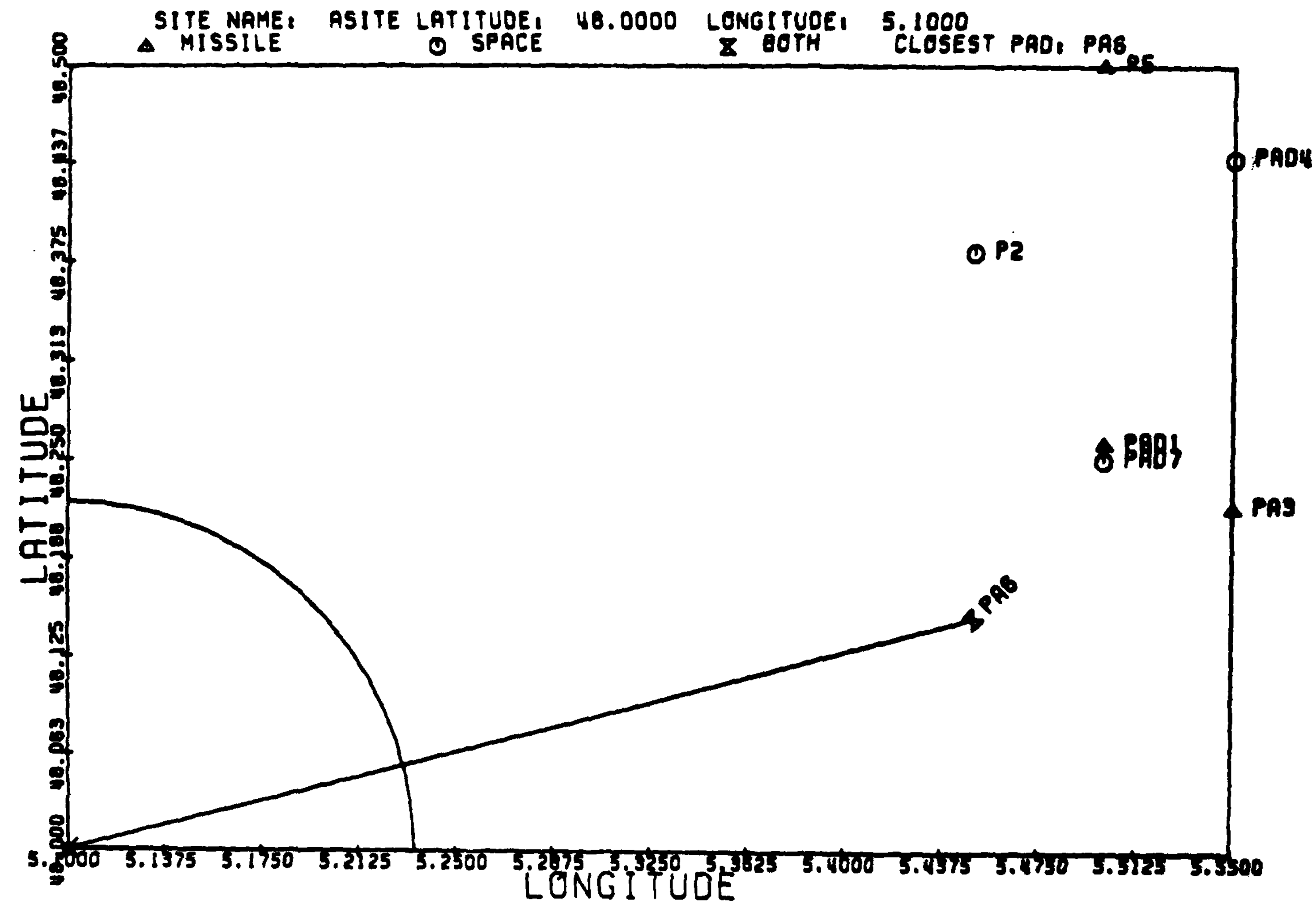


Figure A-89 Example of Zoom on Launch Point Output

may be added to the launch folder by running one of the data base update functions.

A.6.2 Spider Plot Application

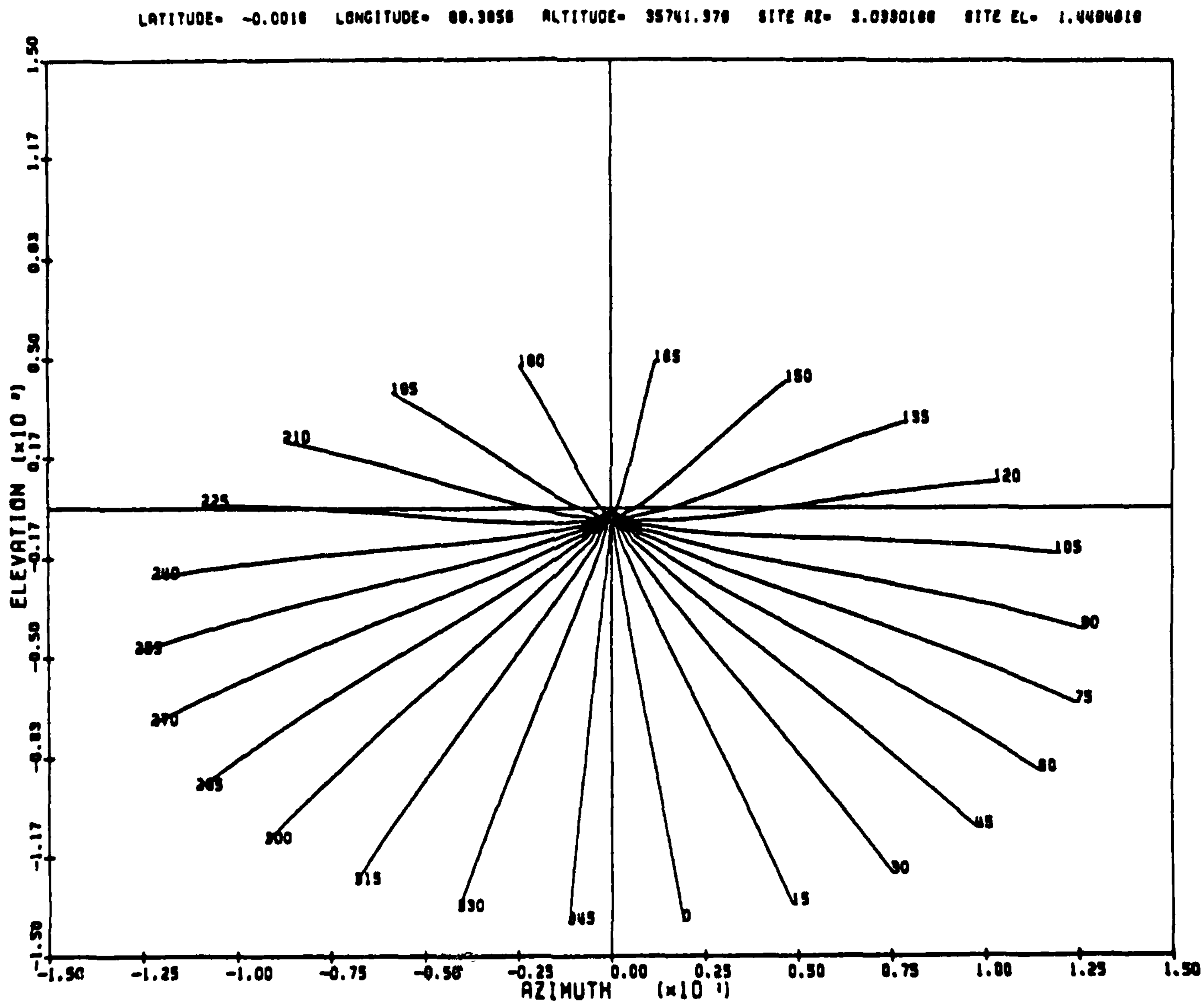
The analyst may wish to generate simulated line-of-sight measurements given a launch point, a sensor location, and an assumed launch vehicle-payload mission pair. Pressing the SPIDER PLOT soft key initiates this application. The application retrieves the launch date, time and position from the current launch folder, and the satellite epoch and orbital element set from the "Blue spaceborne sensor" data base. The analyst is presented with the screen depicted in Figure A-90. The analyst may change the launch identification number. In this case, the application alters the launch date, time and position values. The analyst may change the sensor identification number. In this case, the application alters the satellite epoch and orbital element set. Changing any other field results in the application using the new values in the ensuing calculations. The time vs. intensity and downrange vs. altitude profiles are extracted from the "launch vehicle" data base.

An example of the graphic output is presented in Figure A-91. The elevation vs. azimuth plot shows how the profile appears at various launch azimuths if the event is viewed by the chosen sensor. The radial lines simulate trajectories at 15 degree increments. Given the correct launch point, the correct sensor position, and the correct launch vehicle-payload pair, the analyst may manually superimpose the event line-of-sight observations to estimate the launch azimuth.

```
LAUNCH LOCATION:

      LAUNCH IDENTIFICATION NUMBER: XXXXXXXX
          DATE    : MONTH    : XXXXXXXX    DAY       : XXXXXXXX  YEAR     : XXXXXXXX
          TIME    : HOUR     : XXXXXXXX    MINUTE    : XXXXXXXX  SECOND   : XXXXXXXX
          POSITION: LATITUDE: XXXXXXXX     LONGITUDE: XXXXXXXX   ALTITUDE: XXXXXXXX

SENSOR LOCATION:

      SENSOR IDENTIFICATION NUMBER: XXXXXXXX
               EPOCH:   YEAR: XXXX   DAY: XXX  HOUR: XX  MIN: XX  SECOND: XXXXXXXX
               RIGHT ASCENSION     : XXXXXXXX
               ECCENTRICITY        : XXXXXXXX
               INCLINATION         : XXXXXXXX
               ARGUMENT OF PERIGEE: XXXXXXXX
               MEAN ANOMALY        : XXXXXXXX
               MEAN MOTION         : XXXXXXXX

PROFILE NAME:

      VEHICLE: XXXXXXXX    MISSION: XXXXXXXX
```

Figure A-90 Spider Plot Input Screen

Figure A-91 Example of Spider Plot Output



A.6.3 ALAPP Application

The analyst may wish to estimate a launch plane given only one sensor. Pressing the ALAPP PLOT soft key initiates this application. The application retrieves the launch and sensor identification numbers and the launch vehicle-payload pair from the current launch folder, and presents the screen depicted in Figure A-92. The application retrieves the launch time and position from the "launch folder" data base record for the launch identification number, the sensor epoch and orbital element set from the "Blue spaceborne sensor" data base, the time vs. intensity and downrange vs. altitude profiles from the "launch vehicle" data base, and the IR observations from the "IR inputs" data base for the launch identification number.

An example of the graphic output is presented in Figure A-93. The information screen which accompanies the graphic output is presented in Figure A-94. The intensity vs. time display allows the analyst to check how well the sensor IR data fits the profile curve. If the magnitudes appear correct but shifted in time, the analyst should compare the estimated launch time at the bottom of the information screen against the input launch time. If there is a difference, the analyst may rerun the application with the estimated launch time for input. If the data just does not fit, the analyst may be reasonably sure that the wrong profile was selected. Errors in the first 100 seconds of an event are possible due to atmospheric absorption of the IR data.

The elevation vs. azimuth display is referred to as a partial Spider plot, and is used to check the quality of the fit between the sensor azimuth and elevation data and the projected profile data. The graph is in the true sensor coordinate system. The analyst should verify that the curve through the data points is the same shape as the profile. The calculated launch azimuth is shown as a dotted line. The projected profiles are shown as solid lines on 10 degree increments from the launch azimuth. The junction of all the lines represents the launch site, and the squares are the location of the actual data.

LAUNCH IDENTIFICATION NUMBER: XXXXXXXX

SENSOR IDENTIFICATION NUMBER: XXXXXXXX

PROFILE NAME: LAUNCH VEHICLE: XXXXXXXX PAYLOAD MISSION: XXXXXXXX

Figure A-92 ALAPP Input Screen

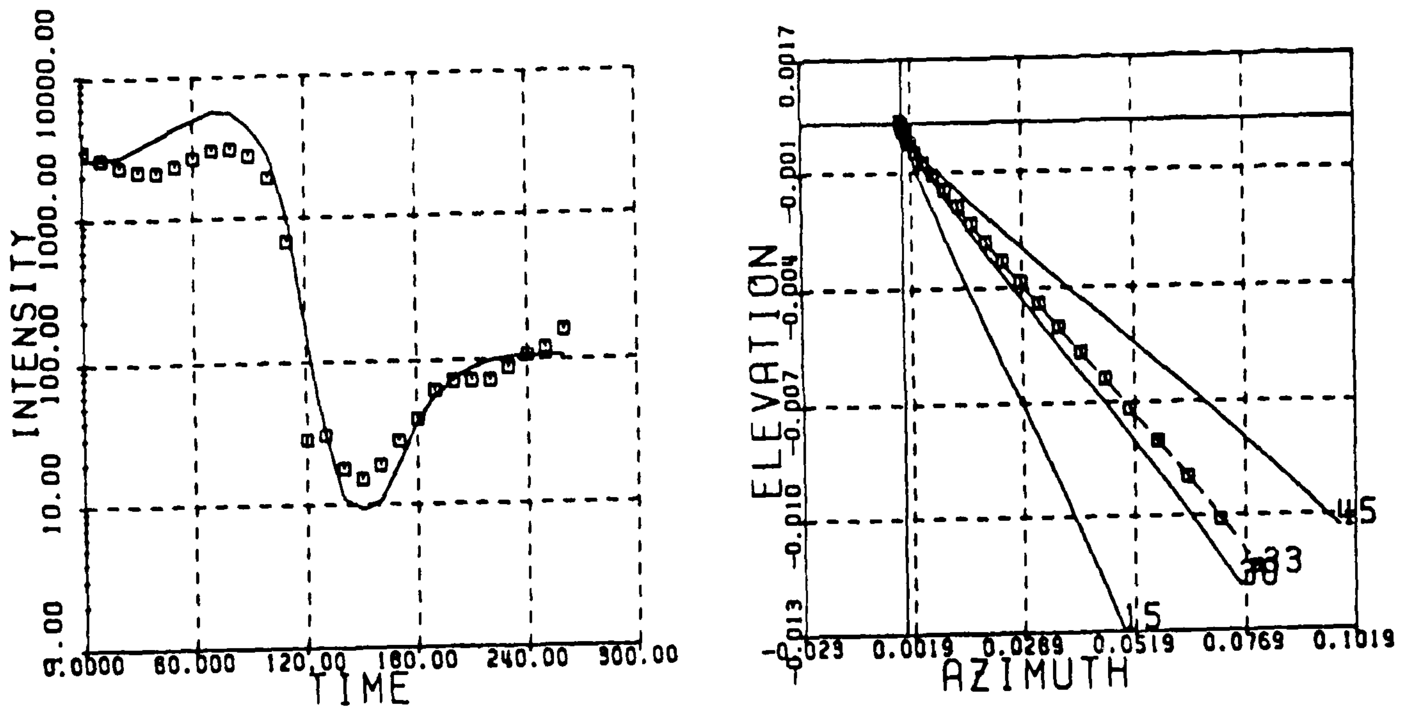


Figure A-93 Example of ALAPP Graphic Output

```
TIME: XX : XX:  XX  DATE: XX :XX : XX
LAUNCH: XXXXXXXX  LAT: XXXXXXXX    LON: XXXXXXXX     ALT:  XXXXXXXX
SENSOR: XXXXXXXX  LAT: XXXXXXXX    LON: XXXXXXXX     ALT:  XXXXXXXX    SNUM: XXXXXXXX
LAUNCH VEHICLE: XXXXXXXX       MISSION:  XXXXXXXX
AZIMUTH: XXXXXXXX       ELEVATION: XXXXXXXX

PROFILE PLOT:                               SPIDER PLOT:
       TIME        INTENSITY                  TIME        AZIMUTH     ELEVATION
    XX  XX  XX      XXXXXXXX                XX  XX  XX    XXXXXXXX    XXXXXXXX
    XX  XX  XX      XXXXXXXX                XX  XX  XX    XXXXXXXX    XXXXXXXX
    XX  XX  XX      XXXXXXXX                XX  XX  XX    XXXXXXXX    XXXXXXXX
    XX  XX  XX      XXXXXXXX                XX  XX  XX    XXXXXXXX    XXXXXXXX
    XX  XX  XX      XXXXXXXX                XX  XX  XX    XXXXXXXX    XXXXXXXX
    XX  XX  XX      XXXXXXXX                XX  XX  XX    XXXXXXXX    XXXXXXXX

                                            ORIGIN:       XXXXXXXX    XXXXXXXX

                                  ESTIMATE     DEVIATION
ESTIMATED LAUNCH TIME             XX : XX : XX  XXXXXXXX
ESTIMATED LAUNCH AZIMUTH          XXXXXXXX     XXXXXXXX
ESTIMATED LAUNCH INCLINATION      XXXXXXXX
INTENSITY ERROR MEASURE                        XXXXXXXX
PEAK INTENSITY % ERROR                         XXXXXXXX
```

Figure A-94 ALAPP Output Screen

The numeric values presented on the information screen give an indication of how well the data matches the displayed profile. This is important since the estimated launch azimuth and launch inclination are based on calculations dependent upon the IR data and the launch information. The smaller the differences are between the estimated and true values, the smaller the standard deviations, the intensity error measure and the peak, intensity per cent error are, the better the data fits the profiles.

A.6.4 Cycle Through ALAPP Application

The analyst may wish to estimate a launch plane given only one sensor for each known launch vehicle-payload mission pair in order to find the best possible fit between the data and stored profiles. Pressing the AUTOMATICALLY CYCLE soft key initiates this application. The application retrieves the launch and sensor identification numbers from the current launch folder, and presents the screen depicted in Figure A-95. The application retrieves the launch date, time and position from the "launch folder" data base record for the launch identification number, the sensor epoch and orbital element set from the "Blue spaceborne sensor" data base, and the IR observations from the "IR inputs" data base for the launch identification number.

The application retrieves the time vs. intensity and downrange vs. altitude profiles for the first (next) launch vehicle-payload mission pair from the "launch vehicle" data base. The calculated graphic and information screen outputs are displayed. At this point, the analyst may request that the application process the profiles for the next launch vehicle-payload mission pair by pressing the RETURN character key, or request that the application terminate by typing in the word "exit". The application exits when all the records of the "launch vehicle" data base have been processed.

An example of the graphic output is presented in Figure A-93. The information screen which accompanies the graphic output is presented in Figure A-94. The intensity vs. time display allows the analyst to check how well the sensor IR data fits the profile curve. If the magnitudes appear correct but shifted in time, the analyst should compare the estimated launch time at the bottom of the information screen against the input launch time. If there is a difference, the analyst may rerun the application with the estimated launch time for input. If the data just does not fit, the analyst may be reasonably sure that the wrong profile was selected. Errors in the first 100 seconds of an event are possible due to atmospheric absorption of the IR data.

LAUNCH IDENTIFICATION NUMBER: XXXXXXXXX

SENSOR IDENTIFICATION NUMBER: XXXXXXXXX

Figure A-95 Cycle Through ALAPP Input Screen

The elevation vs. azimuth display is referred to as a partial Spider plot, and is used to check the quality of the fit between the sensor azimuth and elevation data and the projected profile data. The graph is in the true sensor coordinate system. The analyst should verify that the curve through the data points is the same shape as the profile. The calculated launch azimuth is shown as a dotted line. The projected profiles are shown as solid lines on 10 degree increments from the launch azimuth. The junction of all the lines represents the launch site, and the squares are the location of the actual data.

The numeric values presented on the information screen give an indication of how well the data matches the displayed profile. This is important since the estimated launch azimuth and launch inclination are based on calculations dependent upon the IR data and the launch information. The smaller the differences are between the estimated and true values, the smaller the standard deviations, the intensity error measure and the peak intensity per cent error are, the better the data fits the profiles.

A.6.5 Two Sensor Analysis Application

The analyst may wish to estimate the launch plane for an event observed by two sensors. The advantages of the two sensor method over the one sensor method include the freedom from relying on historical profile data, greater accuracy, and the neglible effects of sensor line-of-sight bias errors on the orbital plane estimation accuracy. Pressing the TWO SENSOR ANALYSIS soft key initiates this application. The application retrieves the identification numbers of the current launch and the two sensors and presents the transaction screen depicted in Figure A-96. The application retrieves the launch time and position from the "launch folder" data base record for the launch identification number, and the epochs and orbital element sets for both sensors from the "Blue spaceborne sensor" data base. It extracts the IR sensor observations for sensor1 from the "IR inputs" data base, and the polynomial coefficients for sensor2 from the "polynomial inputs" data base.

An example of the graphic output is presented in Figure A-97. The information screen which accompanies the graphic output is presented in Figure A-98. The F-G axis display represents the estimated trajectory of the missile within its plane of motion. The East-North display represents the target locations projected onto the East-North horizontal plane of the translating coordinant system. The angle of the fitted straight line with respect to the North axis is the launch azimuth estimate. The crosses on both plots represent the line-of-sight intersections.

The numeric values presented on the information screen provide the analyst with the position and velocity of the missile at burnout, the orbital element set of if the payload, and, the payload will impact the earth, the estimated impact time and location (latitude and longitude).

```
LAUNCH  IDENTIFICATION NUMBER:  XXXXXXXX
SENSOR1 IDENTIFICATION NUMBER:  XXXXXXXX
SENSOR2 IDENTIFICATION NUMBER:  XXXXXXXX
```

Figure A-96 TSATS Input Screen

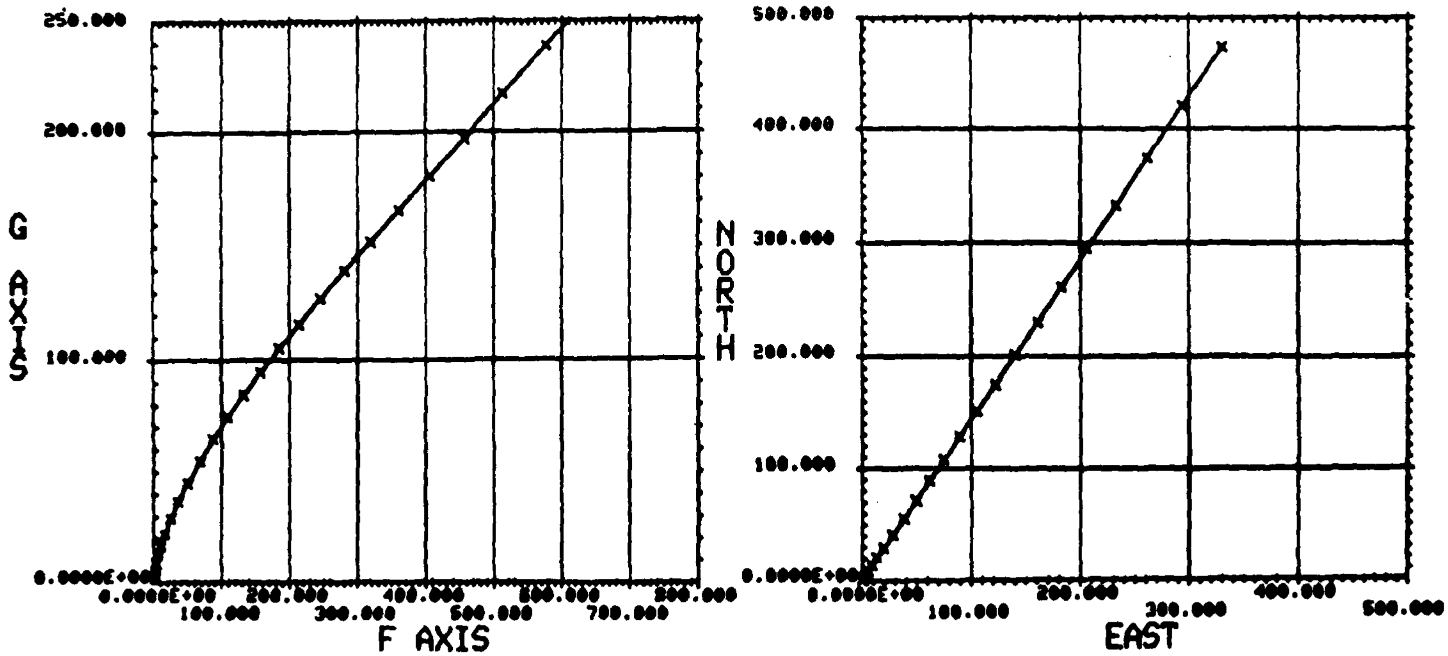


Figure A-97 Example of TSATS Graphic Output

```
TIME: XX : XX : XX  DATE:   XX:XX :XX
LAUNCH:    XXXXXXXX  LAT:   XXXXXXXX  DEG   LON:  XXXXXXXX    DEG   ALT:  XXXXXXXX      M
SENSOR1:   XXXXXXXX  LAT:   XXXXXXXX  DEG   LON:  XXXXXXXX    DEG   ALT:  XXXXXXXX      KM
SENSOR2:   XXXXXXXX  LAT:   XXXXXXXX  DEG   LON:  XXXXXXXX    DEG   ALT:  XXXXXXXX      KM
BILATERATION ANGLE:              DEG

          SENSOR     EPOCH          INC       RT. ASC.    ECC       PERIGEE  M ANOM.   M MOTION
          NUMBER     DAYS           DEG       DEG                   DEG      DEG       REV:DAY

SENSOR1   XXXXXXXX   XXXXXXXX       XXXXXXXX  XXXXXXXX    XXXXXXXX  XXXXXXXX XXXXXXXX  XXXXXXXX
SENSOR2   XXXXXXXX   XXXXXXXX       XXXXXXXX  XXXXXXXX    XXXXXXXX  XXXXXXXX XXXXXXXX  XXXXXXXX

PAYLOAD   XXXXXXXX   XXXXXXXX       XXXXXXXX  XXXXXXXX    XXXXXXXX  XXXXXXXX XXXXXXXX  XXXXXXXX

LAST OBS. TIME: XX : XX: XX   AZIMUTH:  XXXXXXXX    DEG
                                        X=   XXXXXXXX KM   XDOT=   XXXXXXXX     KM:SEC
                                        Y=   XXXXXXXX KM   YDOT=   XXXXXXXX     KM:SEC
                                        Z=   XXXXXXXX KM   ZDOT=   XXXXXXXX     KM:SEC
                                        R=   XXXXXXXX KM   SPEED=  XXXXXXXX     KM:SEC

                  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Figure A-98 TSATS Output Screen

A.7 CONTROL KEY AND SOFT KEY REFERENCE GUIDE

This reference guide provides brief descriptions of the functions of the control keys and programmable soft keys of the S-U 1652 terminal. It is divided into two sections: A.7.1 gives the control keys in alphabetical order; A.7.2 does the same for the programmable soft keys. Following each description is a number in square brackets, "[]", which refers to the section of this manual in which the key is discussed.

A.7.1 Control Keys

Note that the majority of the control keys are not used by the Space and Missile analyst. The descriptions of the unused keys are given for completeness; they are marked with the phrase NOT USED before the description. The user is cautioned not to use these control keys, since they may hinder or damage the operation of SABERS software.

ALARM	NOT USED. Not implemented on the SABERS S-U 1652 terminal. [A.2.4]
BOOT	NOT USED. After initialization, reads the control program from the computer. [A.2.4]
CHAR DEL	Deletes the last character typed. Used during transaction screen editing. [A.2.3]
CLR	Erases text from the monitor on which the cursor appears. Graphics are not affected. Not used during screen editing. [A.2.4]
CNTL	NOT USED. Sends next keyboard character as a control character. [A.2.4]
DEFINE SOFT KEY	NOT USED. Delineates soft key programming mode. [A.2.4]
DOWN ARROW	Moves cursor down to next transaction screen field. Used during transaction screen editing. [A.2.3]
EOF	NOT USED. Sends end-of-file character (CONTROL-Z) to computer. [A.2.4]
ESC	NOT USED. Sends escape character to the computer. [A.2.4]
EXIT	NOT USED. Sends stop-execution character (CONTROL-Y) to computer. [A.2.4]

HOME	Moves cursor to first transaction field on screen. Used during transaction screen editing. [A.2.3]
INHIBIT DISPLAY	NOT USED. Sends suspend-output character (CONTROL-S) to computer. [A.2.4]
INIT	NOT USED. Places terminal in initialized state before RESET or BOOT. [A.2.4]
LEFT ARROW	Moves cursor left to previous transaction field. Used during transaction screen editing. [A.2.3]
LINE DEL	NOT USED. Sends delete-line symbol (CONTROL-U) to computer. [A.2.4]
LOCAL CLEAR GRAPHICS	Clears the graphics buffer and erases the graphics display. Non-graphics text is not affected. [A.2.4]
LOCAL CLEAR SOFT KEYS	NOT USED. Clears all soft key programs. [A.2.4]
NEW SCREEN	Erases the text from, and moves the cursor to, the other monitor. Graphics displays are not affected. Not used during transaction screen editing. [A.2.4]
NEXT FIELD	Advances cursor to the next transaction field. Used during transaction screen editing. [A.2.3]
RELEASE DISPLAY	NOT USED. Sends resume-output symbol (CONTROL-Q) to computer. [A.2.4]
RESET	NOT USED. Returns terminal to initial booted state. [A.2.4]
REVIEW LINE	NOT USED. Sends retype-line symbol (CONTROL-R) to computer. [A.2.4]
RIGHT ARROW	Moves cursor right to next transaction field. Used during transaction screen editing. [A.2.3]

RUBOUT	Deletes the last keyboard character typed. Used during transaction screen editing. [A.2.3]
SEND	Sends the edited transaction screen to the computer. Used during transaction screen editing. [A.2.3]
SHOW SOFT KEYS	NOT USED. Displays all soft key programs on the other monitor. [A.2.4]
TRACE	NOT USED. Shows all terminal interaction. [A.2.4]
UP ARROW	Moves cursor up to previous transaction line. Used during transaction screen editing. [A.2.3]

A.7.2 Programmable Soft Keys

ABORT	Used during transaction screen editing. Terminates both the editing session and the application. [A.2.3.3]
ADD A NEW RECORD	Permits the analyst to add a new record to an existing data base. [A.3.4]
ALAPP PLOT	Enables the analyst to estimate a launch plane given only one sensor. [A.6.3]
AUTOMATICALLY CYCLE	Used to estimate a launch plane given only one sensor for each known launch vehicle-payload pair, in order to find the best fit possible between the data and stored profiles. [A.6.4]
BLUE RADAR SYSTEMS	Allows the analyst to examine the records in the "Blue Radar" data base which match a particular set of search criteria. [A.3.2.2]
BLUE SPACEBORNE SYSTEMS	Allows the analyst to examine the records in the "Blue Spaceborne Sensor" data base which match a particular set of search criteria. [A.3.2.2]
BOTTOM OF PAGE	Used during transaction screen editing. Positions the cursor to the beginning of the first field in the last line of the screen. [A.2.3.3]
CURRENT LAUNCH REVIEW	Allows the analyst to compare the current launch event summary information contained in the current launch folder with historical launch events. (Similar to the launch folder review function.) [A.3.2.6]
DELETE AN EXISTING RECORD	Permits the analyst to delete a record from an existing data base. [A.3.5]

DISPLAY A WORLD MAP	Allows the analyst to draw a world map, and to specify parameters such as projection type, area to be displayed, point above the earth at which the observer is located, and the resolution of the map. [A.4.1.1]
DRAW MAP GRID	Adds a map grid to the current map display. [A.4.1.3]
DRAW POLITICAL BOUNDARIES	Adds political boundaries to the current map display. [A.4.1.2]
ERASE THIS FIELD	Used during transaction screen editing. Results in replacing the content of the field with all blanks and moving the cursor to the beginning of the next field. [A.2.3.3]
EXAMINE CURRENT RECORD	Allows the analyst to reexamine the current record retrieved by the last data base review function. [A.3.2.4]
EXAMINE NEXT RECORD	Allows the analyst to examine the next record retrieved by the last data base review function. [A.3.2.3]
EXAMINE PREVIOUS RECORD	Allows the analyst to examine the record previous to the current record retrieved by the last data base review function. [A.3.2.5]
GENERATE THREAT WINDOWS	Calculates and lists the launch time window and the nominal launch time when a mission may be launched from a launch site with the intention of intercepting a target satellite in its orbit. [A.5.1]
HARDCOPY THIS SCREEN IMAGE	Used during transaction screen editing. Pressing this key at any time during the editing session instructs the system to prepare to automatically print the screen image on the line printer when the SEND control key is pressed. [A.2.3.3]

HARDCOPY TO LINE PRINTER Directs the operating system to prepare to list a file on the line printer. The analyst enters the name of the file to be listed in response to the prompt message "$FILE: ".

INSTRUCTIONS Used during transaction screen editing. Clears the monitor and displays a screen showing the permissable editing features. (See Figure A-4) [A.2.3.3]

IR SENSOR INPUTS Allows the analyst to examine the records in the "IR Inputs" data base which match a particular set of search criteria. [A.3.2.2]

LAUNCH FOLDERS Allows the analyst to examine the records in the "Launch Folder" data base which match a particular set of search criteria. [A.3.2.2]

LAUNCH SITES Allows the analyst to examine the records in the "Launch Site" data base which match a particular set of search criteria. [A.3.2.2]

LAUNCH VEHICLES Allows the analyst to examine the records in the "Launch Vehicle" data base which match a particular set of search criteria. [A.3.2.2]

LIST RADARS VS. ORBIT Calculates and lists the time windows of coverage when each radar in the "Blue radar" data base may observe a space object in its orbit. [A.5.2]

LIST SATELLITE RECONNAISSANCE Calculates and lists the time windows of coverage when each ground facility may be under satellite photo reconnaissance. [A.5.3]

OVERLAY BLUE RADAR COVERAGE For each Blue ground-based sensor contained in the "Blue radar" data base, plots the radar site, the radar site name, and the projection of the radar range along a fan from the minimum azimuth to the maximum azimuth on the current map display. [A.4.2.4]

OVERLAY CURRENT LAUNCH POINT Plots the current launch point and launch identification number (if any) on the map display. [A.4.2.1]

OVERLAY GROUND TRACE Plots the ground trace of a space object in its orbit or of a missile in its trajectory on the current map display. [A.4.2.5]

OVERLAY LAUNCH SITES Plots the launch sites and launch site names (contained in the "launch site" data base) on the current map display. [A.4.2.2]

OVERLAY RADARS VS. ORBIT Plots the location of each Blue ground based sensor which may observe a space object in its orbit. [A.4.2.7]

OVERLAY RED SUPPORT FACILITIES Plots the location and name of each Red tracking and support facility (contained in the "tracking facilities" data base) on the current map display. [A.4.2.3]

OVERLAY SATELLITE RECONNAISSANCE Plots the area on the earth's surface observed by a camera mounted on a photo reconnaissance satellite. This is done by displaying the intersection of the cone originating at the camera with the earth at equal time-spaced points of the satellite in its orbit. [A.4.2.8]

OVERLAY TIME MARKS ON GROUND TRACE Allows the analyst to visualize the time spent by a space object in each part of its orbit, by superposing equal time-spaced tic marks on the ground trace of a space object in its orbit. [A.4.2.6]

POLYNOMIAL INPUTS Allows the analyst to examine the records in the "Polynomial Inputs" data base which match a particular set of search criteria. [A.3.2.2]

PRINT LAST SCREEN IMAGE Used only when not engaged in transaction screen editing. Causes the last screen image displayed to be printed on the line printer. [A.1.4]

RADAR INPUTS	Allows the analyst to examine the records in the "Ground Based Sensor Inputs" data base which match a particular set of search criteria. [A.3.2.2]
RADARS VS. ORBIT	Combines listing-generating and graphics overlay output. Calculates and lists the time windows of coverage when each radar in the "Blue radar" data base may observe a space object in its orbit, and plots each radar which may observe the object. [A.5.4.1]
RED SUPPORT FACILITIES	Allows the analyst to examine the records in the "Tracking Facilities" data base which match a particular set of search criteria. [A.3.2.2]
REDRAW MAP ONLY	Results in the current graphic display being erased, and the map only being drawn again according to the parameters used the last time the DISPLAY A WORLD MAP function was used. [A.4.1.4]
RETYPE THE SCREEN	Used during transaction screen editing. Results in the clearing of the monitor and redisplaying the screen. [A.2.3.3]
SATELLITE RECONAISSANCE	Combines listing-generating and graphics overlay output. Calculates and lists the time windows of coverage when each ground facility may be under satellite photo reconnaissance, and displays the intersection of the cone originating at the camera with the earth at equal time-spaced points of the satellite in its orbit. [A.5.4.2]
SELECT LAUNCH ID	Initiates setting the default launch identification number. [A.3.3.2]
SELECT PAYLOAD ID	Initiates setting the default payload identification number. [A.3.3.2]
SOVIET SOB	Allows the analyst to examine the records in the "Soviet ESV Status" data base which match a particular set of search criteria. [A.3.2.2]

SPIDER PLOT	Generates simulated line-of-sight measurements, given a launch point, a sensor location, and an assumed launch vehicle-payload mission pair. [A.6.2]
SUMMARY	Provides a line printer listing of all the records in a data base which satisfy the search criteria entered by the analyst. [A.3.2.7]
TOP OF PAGE	Used during transaction screen editing. Positions the cursor to the beginning of the first field on the first line of the screen. [A.2.3.3]
TWO SENSOR ANALYSIS	Estimates the launch plane for an event observed by two sensors. [A.6.5]
UPDATE AN EXISTING RECORD	Allows the analyst to add to or to correct an existing data base record. [A.3.3.1]
VIEW AT TERMINAL	Directs the operating system to prepare to list a file on the terminal. The analyst enters the name of the file to be listed in response to the prompt message "$FILE: ".
ZOOM ON LAUNCH SITE	Initiates application to determine which launch site, and which launch pad of the site, are the true launch position of a given launch event. [A.6.1]

A.8 ERROR MESSAGES

Whenever an analyst deals with data using a computer, errors are bound to occur. In SABERS, many different error situations are possible because of the complexity of the system. However, many of these error situations should only occur during application development, and not as a result of analyst-machine interaction. The purpose of this section is to identify the errors that are a result of something the analyst has control over. Any error messages received during the analyst's duty on the watch that are not presented in this section should be reported to the system manager as soon as possible, since they may be an indication of system malfunction or corruption.

Consistent error handling is one of the design considerations of SABERS. The analyst should be informed of the existence of an error condition. The application should present blank fields for input if the error hinders creating default information. The application may exit with an error message if the error hinders output, or may attempt to recover, depending on the type and severity of the error.

The error messages are presented in two sections. The first section lists the errors that may occur during transaction processing, and the second section lists the errors that may occur during application execution.

A.8.1 Transaction Processing Error Messages

Message: "INPUT NUMBER OUT OF RANGE"
Action: Edit the field flagged with blinking question marks with a number within the required limits.

Message: "INPUT NOT PROPER INTEGER"
Action: Edit the field flagged with blinking question marks with a properly constructed integer value.

Message: "INPUT NOT ONE OF THE SELECT OPTIONS"
Action: Edit the field flagged with blinking question marks with a value from the specified choices.

Message: "DISALLOWED INPUT TEXT CHARACTER"
Action: Edit the field flagged with blinking question marks using only acceptable characters.

Message: "IMPROPER REAL NUMBER CONSTRUCTION"
Action: Edit the field flagged with blinking question marks with a properly constructed real number.

Message: "MANDATORY INPUT FIELD"
Action: Edit the field flagged with blinking question marks with an appropriate value, since the field may not be left blank.

Message: "SORRY. THAT WAS AN ILLEGAL CHARACTER"
Action: Do not type the illegal character again.

A.8.2 Application Execution Errors

Message: "NEEDED DATA NOT PROVIDED"
Cause: Application expects data in a blank field.
Action: Edit the field flagged with blinking question marks with an appropriate value, since the field may not be left blank.

Message: "NEEDED DATA NOT GIVEN"
Cause: Application expects data in a blank field.
Action: Edit the field flagged with blinking question marks with an appropriate value, since the field may not be left blank.

Message: "THIS IS A MANDATORY INPUT FIELD"
Cause: Application expects data in a blank field.
Action: Edit the field flagged with blinking question marks with an appropriate value, since the field may not be left blank.

Message: "SYNTAX ERROR IN INPUT STRING"
Cause: Application finds incorrect syntax in the specification of an assertion.
Action: Edit the field flagged with blinking question marks with a properly constructed assertion.

Message: "THERE ARE NO RECORDS MATCHING THE CONDITIONS"
Cause: Application cannot find any records in the data base matching the search criteria.
Action: Change the search criteria.

Message: "RECORD # XXX OF YOUR LAST REVIEW HAS BEEN DELETED"
Cause: A data base record has been deleted by another analyst since you retrieved it.
Action: None.

Message: "NO LAUNCH SITE WITHIN LAT AND LON RANGE"
Cause: Application cannot find a launch site within the specified latitude and longitude range from the input reported launch position.
Action: Either use larger ranges or a different reported launch position.

Message: "THERE ARE NO LAUNCH FOLDERS MATCHING THE CONDITIONS SPECIFIED"
Cause: Application cannot find any records in the "launch folder" data base matching the search criteria.
Action: Change the search criteria.

Message: "NO RECORD WITH LAUNCH ID NUMBER XXXXXX"
Cause: Application cannot find a record in the "launch folder" data base with the input launch identification number.
Action: Enter a new launch identification number, or fill in all the blank fields manually.

Message: "USING LAUNCHID FOUND NO RECORD IN LAUNCHFOLDER FILE"
Cause: Application cannot find a record in the "launch folder" data base with the input launch identification number.
Action: Enter a new launch identification number, or fill in all the blank fields manually.

Message: "FOUND XXXXX RECORDS WITH LAUNCH ID NUMBER XXXXXX"
Cause: Application found more than one launch folder for the input launch identification number.
Action: The launch identification number should be unique. Delete all but one of the records with the duplicate launch identification number in the "launch folder" data base.

Message: "USING LAUNCHID FOUND >1 RECORD IN LAUNCHFOLDER FILE"
Cause: Application found more than one launch folder for the input launch identification number.
Action: The launch identification number should be unique. Delete all but one of the records with the duplicate launch identification number in the "launch folder" data base.

Message: "BAD DATA IN LAUNCHFOLDER FILE"
Cause: Application detected data in the "launch folder" data base that is out of range.
Action: Check the launch date, time and position for that record in the data base.

Message: "NO BLUE GROUND RECORDS FOUND"
Cause: All the "Blue ground based sensor" data base records have been deleted.
Action: None.

Message: "NO RECORD IN GROUND INPUTS FILE FOR OBJECT ID XXXXXX"
Cause: Application cannot find a record in the "ground based sensor inputs" data base with the input identification number of the object being observed.
Action: Enter a new satellite identification number, or fill in all the blank fields manually.

Message: "NO RECORD IN GROUND INPUTS FILE FOR OBJECT ID XXXXXX WITH EPOCH TIME LESS THAN START TIME"
Cause: Application cannot find a record in the "ground based sensor inputs" data base with the epoch time less than the start time.
Action: Enter a later start time, or fill in all the blank fields manually.

Message: "NO RECORD WITH LAUNCH SITE NAME XXXXXXXX"
Cause: Application cannot find a record in the "launch site" file for this launch site name.
Action: Enter a new launch site name, or fill in all the blanks manually.

Message: "USING SENSORID FOUND NO RECORD IN BLUESPACE FILE"
Cause: Application cannot find a record in the "Blue spaceborne sensor" data base with the input sensor identification number.
Action: Enter a new sensor identification number, or fill in all the blank fields manually.

Message: "BAD DATA IN BLUESPACE FILE"
Cause: Application detected data in the "Blue spaceborne sensor" data base that is out of range.
Action: Check the epoch and orbital element set data for that record in the data base.

Message: "USING LAUNCHVE FOUND NO RECORD IN LAUNCHVEHICLES FILE"
Cause: Application cannot find a record in the "launch vehicle" data base with the input launch vehicle name-payload mission name pair.
Action: Add the profile model for this pair to the data base, or enter a new launch vehicle-payload mission pair.

Message: "USING LAUNCHVE FOUND >1 RECORD IN LAUNCHVEHICLES FILE"
Cause: Application found more than one profile model for the input launch vehicle name-payload mission name pair.
Action: The profile model should be unique. Delete all but one of the records with the the duplicate launch vehicle-payload mission pair in the "launch vehicle" data base.

Message: "USING LAUNCHID FOUND NO RECORD IN IRINPUTS FILE"
Cause: Application cannot find a record in the "IR inputs" data base with the input launch identification number.
Action: Add the IR sensor data for this launch identification number to the data base, or enter a new launch identification number.

Message: "USING LAUNCHID FOUND >1 RECORD IN IRINPUTS FILE"
Cause: Application found more than one record of IR input data for the input launch identification number.
Action: The IR inputs should be unique. Delete all but one of the records with the duplicate launch identification number in the "IR inputs" data base.

Message: "USING LAUNCHID FOUND NO RECORD IN POLYNOMIAL FILE"
Cause: Application cannot find a record in the "polynomial inputs" data base with the input launch identification number.
Action: Add the polynomial data for this launch identification number, or enter a new launch identification number.

Message: "USING LAUNCHID FOUND >1 RECORD IN POLYNOMIAL FILE"
Cause: Application found more than one record of polynomial input data for the input launch identification number.
Action: The polynomial inputs should be unique. Delete all but one of the records with the duplicate launch identification number in the "polynomial inputs" data base.

Message: "OELECI FAILS TO CONVERGE FOR DT = XXXXX.XXX"
Cause: Newton iteration fails to converge for the given orbital element set while calculating the eccentric anomaly.
Action: Expect that any output results of the application are invalid for this time. Check that the sensor orbital element set is reasonable. Also, check the sensor date and time so that the perturbation is not over a long period of time.

Message: "RECORDS FOUND= XXXX RECORDS DISPLAYED= XXXX"
Cause: More facilities exist than may be displayed at one time.
Action: Realize that not all facilities have been displayed.

Message: "ALAPP -> SENLOC SIAE ECCENT. (ONE) INVALID"
Cause: The eccentricity contained in the "Blue spaceborne sensor" data base orbital element set is out of range.
Action: Check the eccentricity in the data base for the Blue spaceborne sensor identification number.

Message: "ALAPP -> SENLOC SPAM MOTION (ZERO) INVALID"
Cause: The mean motion contained in the "Blue spaceborne sensor" data base orbital element set is out of range.
Action: Check the mean motion in the data base for the Blue spaceborne sensor identification number.

Message: "ALAPP -> SENLOC SENSOR POSITION UNOBTAINABLE"
Cause: Newton iteration fails to converge for the given orbital element set in the "Blue spaceborne sensor" data base while calculating the eccentric anomaly.
Action: Expect that any output results of the application are invalid for this time. Check that the sensor orbital element set for the Blue spaceborne identification number is reasonable. Also, check the sensor date and time so that the perturbation is not over a long period of time.

Message: "PCYCLE -> SENLOC SIAE ECCENT. (ONE) INVALID"
Cause: The eccentricity contained in the "Blue spaceborne sensor" data base orbital element set is out of range.
Action: Check the eccentricity in the data base for the Blue spaceborne sensor identification number.

Message: "PCYCLE -> SENLOC SPAM MOTION (ZERO) INVALID"
Cause: The mean motion contained in the "Blue spaceborne sensor" data base orbital element set is out of range.
Action: Check the mean motion in the data base for the Blue spaceborne sensor identification number.

Message: "PCYCLE -> SENLOC SENSOR POSITION UNOBTAINABLE"
Cause: Newton iteration fails to converge for the given orbital element set in the "Blue spaceborne sensor" data base while calculating the eccentric anomaly.
Action: Expect that any output results of the application are invalid for this time. Check that the sensor orbital element set for the Blue spaceborne identification number is reasonable. Also, check the sensor date and time so that the perturbation is not over a long period of time.

Message: "SPICMD -> SENLOC SIAE ECCENT. (ONE) INVALID"
Cause: The eccentricity contained in the "Blue spaceborne sensor" data base orbital element set is out of range.
Action: Check the eccentricity in the data base for the Blue spaceborne sensor identification number.

Message: "SPICMD -> SENLOC SPAM MOTION (ZERO) INVALID"
Cause: The mean motion contained in the "Blue spaceborne sensor" data base orbital element set is out of range.
Action: Check the mean motion in the data base for the Blue spaceborne sensor identification number.

Message: "SPICMD -> SENLOC SENSOR POSITION UNOBTAINABLE"
Cause: Newton iteration fails to converge for the given orbital element set in the "Blue spaceborne sensor" data base while calculating the eccentric anomaly.
Action: Expect that any output results of the application are invalid for this time. Check that the sensor orbital element set for the Blue spaceborne identification number is reasonable. Also, check the sensor date and time so that the perturbation is not over a long period of time.

Message: "STAGE INTENS. MISSING IN PROFILE"
Cause: Application cannot find the stage intensity in the altitude (32) field in the "launch vehicle" data base.
Action: Enter the maximum intensity for the launch vehicle-payload mission pair.

Message: "ALAPP -> AZMTH NO SOLUTION CHECK INPUT DATA SET"
Cause: No intersection exists between the line-of-sight and the profile.
Action: Check the launch position in the "launch folder" data base for the launch identification number and the orbital element set of the sensor in the "Blue spaceborne sensor" data base.

Message: "PCYCLE -> AZMTH NO SOLUTION CHECK INPUT DATA SET"
Cause: No intersection exists between the line-of-sight and the profile.
Action: Check the launch position in the "launch folder" data base for the launch identification number and the orbital element set of the sensor in the "Blue spaceborne sensor" data base.

Message: "->PSEN ->DATPRO ->LINEX LINE OF SIGHT ERROR, CHECK DATA"
Cause: The lines-of-sight of both sensors are parallel.
Action: Check that two unique sensors are being used.

Message: "->PSEN ->DATPRO NOT ENOUGH DATA POINTS (<4)"
Cause: Application needs at least four points of sensor observations.
Action: Check the "IR inputs" data base for the launch identification number.

Message: "->PSEN ->DATPRO TIME TAGS OUT OF SEQUENCE"
Cause: Application needs the time tags to be received in chronological order.
Action: Check the "IR inputs" data base for the launch identification number.

Message: "->PSEN ->DATPRO ->ORBELS CIRCULAR ORBIT, PERIGEE UNDEFINED"
Cause: Application determines that the payload has a circular orbit.
Action: Expect errors in the estimates.

Message: "->PSEN ->DATPRO ->ORBELS EQUITORIAL ORBIT: NODE UNDEFINED"
Cause: Application determines that the payload has an equatorial orbit.
Action: Expect errors in the estimates.

Message: "->PSEN ->DATPRO ->ORBELS ALTITUDE TOO LOW: < 50 KM."
Cause: The burnout altitude of the rocket is less than 50 kilometers.
Action: Expect errors in the impact point prediction.

Message: "->DATPRO ->IMPRED INACCURATE PREDICTION"
Cause: Newton iteration diverged while calculating the target orbital elements.
Action: Expect errors in the estimates.

Message: "->SENECI-> DATPRO ->KALMAN MATRIX INVERSION ERROR"
Cause: Application detected the inversion of a singular matrix.
Action: Expect that any output results of the application are invalid for this time. Check that the sensor orbital element set for the Blue spaceborne identification number is reasonable. Also, check the sensor date and time so that the perturbation is not over a long period of time.

Message: "->PSEN ->SENECI SENSOR POSITION UNOBTAINABLE"
Cause: Newton iteration fails to converge for the given orbital element set

in the "Blue spaceborne sensor" data base while calculating the eccentric anomaly.

Action: Expect that any output results of the application are invalid for this time. Check that the sensor orbital element set for the Blue spaceborne identification number is reasonable. Also, check the sensor date and time so that the perturbation is not over a long period of time.

APPENDIX B

APPLICATIONS PROGRAMMER MANUAL AND PROGRAM MAINTENANCE REFERENCE MANUAL FOR THE SABERS MAP DRAWING CAPABILITY

CONTENTS

LIST OF FIGURES

B.1 TITP APPLICATION PROGRAMMER MANUAL

The subsections which follow describe in detail the use of the routines which make up the SABERS Terminal Independent Transaction Processor. Two general types of routines are included: stand-alone utilities for the creation and maintenance of the SABERS screen images; and subroutines used in the manipulation of those screen images for user interaction. The routines are presented in alphabetical order as follows:

B.1.1 CRE8

User input to SABERS is through the Terminal Independent Transaction Processor (TITP). This is a fill-in-the-blanks interaction in which the user is shown a form on the screen and is expected to fill in the form with the data which will then control the action of the SABERS-supported application program (henceforth, simply the application). The software tool which enables an application programmer to design and build the form to be presented, and also to define the legal inputs for each blank in the form, is called CRE8. CRE8 is actually a small compiler which takes as its source code the text of a screen image definition and produces an efficient, compact internal form which is required for TITP.

The technique for defining screen images in the screen image definition language is such that fields may appear anywhere on the screen and may be rectangular, not just a single line. Further flexibility is obtained by permitting groups of fields to be subconstituents of enclosing fields. In fact, for most considerations the screen image as a whole is itself treated as just a field.

The screen image designer can determine whether changes may be made to individual fields at run time. For those fields which permit input, the screen definition spells out the legal data types and values.

B.1.1.1 Screen Image Definition

The first piece of information in a screen image definition is the name chosen for the screen image. This name must not be the name of an existing screen image. The name is composed of any characters except blank or colon. It may be up to 80 characters long although fewer are usually more practical. The screen image name, as is true for all field names, must be followed by a colon to separate it from the rest of the definition. CRE8 will map the name into a system file name with exactly eight characters, and composed of

alphanumerics only.

Once the specifications and subfields for one screen image are complete, other screen image definitions may follow to the actual end of the input file. An example screen image definition may be seen in Figure B-1. The resulting screen image appears in Figure B-2.

B.1.1.2 Fields

The whole screen image is defined in rectangular pieces called "fields." The screen may be considered a single large field with other fields nested within it. The fields defined within a screen image appear between angle brackets and are separated by semicolons.

B.1.1.2.1 Field Names

Field names have the same characteristics and restrictions as screen image names as described above. The first part of a field definition must be its name, and that name must be followed by a colon.

B.1.1.2.2 Field Extents

The extent of a field, the definition of the area of the screen it will cover, is specified by four numbers placed within parentheses. The numbers represent, respectively, the lowest numbered row, followed by a colon; the highest numbered row, followed by a comma; the lowest numbered column, followed by a colon and the highest numbered column to be included in the field. These row and column numbers are counted with respect to the enclosing field. That is, a field of one row and one column located on the screen at row #50 and column #22 may contain one and only one subfield and that subfield may only have the extent (1:1,1:1), because the subfield is counted relative to the start of the enclosing or next outer field. The syntax for specifying an extent is:

```
        (* THIS IS AN EXAMPLE OF A SABERS SCREEN IMAGE DEFINITION *)
        (* THESE LINES ARE RECOGNIZED AS COMMENT LINES *)

AUTO: (7:19,1:55)NOU
        'VEHICLE ID NUMBER:/',
        'MAKE:/',
        'MODEL:/',
        'YEAR:[NL2]',
        'PURCHASER INFORMATION -'
<
        ID#: (1:1,20:40) FREE [!ADT_];
        MAKE: (2:2,7:27) FREE [S'PLYMOUTH','CHEVROLET','FORD'];
        MODEL: (3:3,8:28) FREE [ADT_];
        YEAR: (4:4,7:10) FREE [I1900:2000];
        PRCHINFO: (7:13,5:55)
        'NAME:/',
        'STREET:/',
        'CITY:/',
        'STATE:[NL2]',
        'PURCHASE DATE -'
        <
                NAME: (1:1,7:36) NOA [ATS_];
                STREET: (2:2,9:48) FREE [ADTS_];
                CITY: (3:3,7:26) FREE [AT_];
                STATE: (4:4,8:17) FREE [AT_];
                PRCHDATE: (7:7,5:32)
                'MONTH:    DAY:    YEAR:'
                <
                        MONTH: (1:1,8:9) FREE [I1:12];
                        DAY: (1:1,16:17) FREE [I1:31];
                        PYEAR: (1:1,25:28) FREE [I1900:2000]
                >
        >
>
```

Figure B-1 Screen Image Definition Example

```
VEHICLE ID NUMBER: XXXXXXXXXXXXXXXXXXXXX
MAKE: XXXXXXXXXXXXXXXXXXXXX
MODEL: XXXXXXXXXXXXXXXXXXXXX
YEAR: XXXX

PURCHASER INFORMATION -
    NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    STREET: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    CITY: XXXXXXXXXXXXXXXXXXXX
    STATE: XXXXXXXXXX

    PURCHASE DATE -
        MONTH: XX DAY: XX YEAR: XXXX
```

Figure B-2 Example Screen Image

(row-low : row-high , col-low : col-high)

Because of common screen sizes, a typical Screen Image extent is "(1:24, 1:80)".

A subfield is not allowed to extend beyond the bounds of its enclosing field. A similar rule, dictated by TITP's inability to interpret a cell which belongs to two input fields, is that field extents which define overlapping field areas are also not permitted. A field may be multi-row as well as multi-column, which enhances the descriptive power of the screen definition language.

CRE8 will automatically duplicate the low value when the colon and high value are left out of the extent. Thus the example could have been written "(1,1)". An extent is mandatory for each field.

B.1.1.2.3 Input Access

Some fields are for text which is never changed during a run, some are to be changed only by the application user, some are to be changed only by the application program itself, and some fields may admit input from either the user or the application. It is important for automatic error checking at run time, and to enable the cursor to be limited to user input (so-called "unprotected") fields, that these access types be specified for each field. The four keywords used to supply this information to CRE8 are:

LOCK	no input from the user or the application: this field is constant
NOA	no input from the application is permitted: this is user input only
NOU	no input from the user, application input only; this is the default
FREE	input permitted from either user or application

The access specified for a field takes precedence over that of the enclosing field, but only in the area covered by the enclosed field. The example will clarify this. If no access is specified, the current field inherits the access status of its enclosing field.

B.1.1.2.4 Input Constraints

Fields whose definition permits input must define what inputs are legal so that TITP can check for invalid inputs at run time. The types of input permitted for any application using TITP include real numbers, integers, character strings and selected words. The number of columns implied by the input specification may not exceed the field width as defined in the extent. Ranges are specified as lower bound followed by upper bound. This order is required. The default values for the range are -1E37 and 1E37 respectively.

Real numbers may take a variety of forms on input according to their use. The application designer may choose either the ordinary form which uses simple signed numbers with decimal part, like -25.63 or the scientific (computer-linearized) form using a simple real, "E", and a power of ten, for example 6.02E23. In either case, a range of acceptable values (also called bounds) is part of the definition. For a real number, the range itself may be written in either form. The choice of type is given by the letter R or E followed by the bound values, which are separated by a colon. The time of day might be constrained, using hours and hundredths, as "[R0.00:24.00]" or "[R.00E0:2.4E1]" with equal results.

Integers specified with the letter I and the range, which is indicated by ı pair of integers separated by colons. Input of an inventory value might ıave the definition "[I0:999]".

Character Strings have the permissions of alphabetic ("A"), numeric "D"), blanks ("_"), trailing blanks ("T") or special characters ("&"). The ɜy letters are used in combination to limit the classes of character input ɜrmitted. For a field to contain only U.S. State names, "[A]" would suffice ıile a field for people's names, which can contain special characters and .anks (like M'Butu J. Hartford-Smythe) would require "[A&_]." A computer me field might require "[AD&]."

Selected words limit the input to a specific list of words of any individual configuration. The indicating letter is S. It is followed by a list of quoted words separated by commas. Such a list might be "[S'FORD','CHEVY','BUICK']". Input to a selected word field must exactly match one of the listed items.

B.1.1.2.5 Initialization

All of the text which is visible on the screen when it is first seen at run time is put there through the initialization facility. In its simplest form, the initialization is represented by a quoted string associated with the field within which that string is to appear.

The default value for each character position in a screen image is blank. There are several methods for making certain common forms easy to define. Suppose the heading for a form is to read "FORM A-37". Assuming the heading is part of the outermost field, the screen image, and the extent is "(1:24,1:80)", the init is simply

```
"            FORM A-37".
```

The twelve blanks may be included in the init by using the format directive "[BL 12]".

The available format directives are

[BL n] one or more (n) blanks to be inserted, e.g. "[BL 3]"
[NL n] one or more (n) new lines to be inserted, e.g. "[NL 3]"

The linear initialization string is written into the field rectangle by automatically wrapping around to the start of the next line when necessary. The field is filled only when all of the last line is filled or when a newline symbol is encountered while on the last line of the field.

Since the [] characters are used to delimit format directives, they may not appear as themselves in a string. If brackets are to be included, the following method may be used. Suppose we wish A[3;5] to be the initialization text. We would have to define an init string 'A['['] 3;5 [']']'.

B.1.1.2.6 Subfields

As noted above, any field may have within it a group of subfields. The subfields appear separated by semicolons, enclosed in angle brackets. An example, simplified by substituting the symbol * for the field characteristics, might be:

```
SINAME: *
<FIELD1: *; FIELD2: *>
```

This screen image, named "SINAME" contains two subfields. These subfields, or "offspring", both have the same "parent" field and so are called "siblings". These can be further refined with additional nesting:

```
SINAME: *
<FIELD1: * <F11: *; F12: * <F111: *; F112:8> ; F13: * > ;
FIELD2: * <F21: *>  >
```

Here, SINAME contains two fields, FIELD1 contains three fields, F12 contains two fields, and FIELD2 contains a single field.

B.1.1.3 Language Syntax

The complete syntax of the screen definition language is presented in Section B.2.1.4.

B.1.1.4 Relating Screen Image to Transaction Processing

Initialization is automatically placed on the resulting screen image as controlled by the field nesting and the extent of the field it is associated with.

The input fields are "unprotected" at run time; that is, they are areas into which the user may write. The TITP processing will automatically constrain the terminal cursor to the unprotected areas, so that the user cannot change any of the items outside of the input field areas.

The input values are automatically checked by TITP for conformity to the specifications in the screen image definition and will not allow the application to see any of the input on the form until all input fields are found to be valid.

For some applications, there are certain fields for which user input is required. This can be indicated by using the "mandatory input" symbol as the first character within the input constraint for that field. The symbol is the "!" character, meaning that input is required.

B.1.1.5 Messages

The following messages may be generated during execution of CRE8:

Messages	Probable Cause or Meaning
ACCESS - Premature End of File	missing ">"
ACCESS - Specification Error	incorrect access word
<<End of Run>>	normal termination
EXTENT - Missing Specification	missing "(" likely
EXTENT - Missing Col Specification	missing ")" likely
EXTENT - Range Overflow	number more than 4 digits
EXTENT - Bad Character	
EXTENT - Bound Exceeds Enclosing Bounds	
FLDNAM - Duplicate Field Name	duplicates a sibling; use a different name
GETBND - "E" Not Allowed Here	incorrect form for an input constraint range element
GETBND - "." Not Allowed Here	incorrect form for an input constraint range element
GETBND - "+" Not Allowed Here	incorrect form for an input constraint range element
GETBND - "-" Not Allowed Here	incorrect form for an input constraint range element
INIT - Record Overflow	NL or "/" symbol on bottom line of field
INIT - Field Overflow	text exceeds field limits
SPECS - Premature E-O-F	missing field closure symbol(s) ">"
SPECS - Bad Segment Type After Extent	must be access, initialization, constraint or subfield
SPECS - Bad Segment Type After Access	must be initialization, constraint or subfield
SPECS - Bad Segment Type After Init	must be constraint or subfield
SPECS - Bad Segment Type After Rule (Constraint)	must be subfield or field end
CONSTR - User Input Not Allowed By Access	! for mandatory input only on FREE or NOA
CONSTR - Bad Character, Not Allowed By Access	select strings must be single quoted
EXT - + Not Allowed Here	bad form for extent value
EXT - - Not Allowed Here	bad form for extent value
EXT - Non-digit Or Overflow	bad form for extent value

B.1.2 DELSI

DELSI is a stand-alone maintenance routine for the transaction processor. Its purpose is the removal of an outdated screen image from the permanent SABERS library. The following example illustrates the use of this routine (the user input is capitalized).

```
$ DELSI
Which screen image should be deleted?
AUTO
Now delete files ZWGZAZZF.*;*
$ DEL ZWGZAZZF.*;*
```

B.1.3 EDIT

EDIT is the subroutine which allows a SABERS application to insert values into a screen image prior to displaying the screen to the user. The routine permits editing of a single field, any logical group of fields, or the entire screen image with a single call. A screen image edit is initiated by a subroutine call in the following form:

```
CALL EDIT (QNAME, BUFFER)
```

QNAME represents the qualified name of a screen image, e.g. 'AUTO'; a screen image and a group of fields, e.g. 'AUTO:PRCHINFO'; or a single field, e.g. 'AUTO:PRCHINFO:PRCHDATE:PYEAR' . The QNAME should be a string terminated by a null byte, such as a FORTRAN literal. Screen image and field names within QNAME must be separated by colons, and the QNAME is enclosed in single quotes.

BUFFER is a value or series of values which will be inserted into the screen image. The values may be of data type integer, real or character. There is no need for the application program to encode numeric values to ASCII format; EDIT will perform this task. When a single field is to be edited, BUFFER should simply be the value to be inserted. However, when multiple fields are to be edited, the values must be in a contiguous series of bytes in an order corresponding to the screen image's logical structure. This contiguity of values may be attained in two ways: by the use of an all-encompassing byte array whose components are equivalenced with the values to be inserted; or by declaring a common area in which the values are arranged in their proper order. In the first case the parameter BUFFER would be the name of the equivalenced array. In the second, BUFFER would be the name of the first variable in the common area.

For example, to EDIT the purchase date portion of our AUTO screen image:

```
INTEGER*2 MONTH, DAY, YEAR

COMMON /PDATE/ MONTH, DAY, YEAR

MONTH = 10
DAY = 8
YEAR = 1975

CALL EDIT ('AUTO:PRCHINFO:PRCHDATE', MONTH)
```

The following application-oriented error messages may be generated by EDIT and are self-explanatory:

"The field specified by the QNAME: ________ is locked from editing by the application program."
"The field specified by the QNAME: ________ does not exist."
"The screen image requested by the QNAME: ________ could not be found in either permanent or temporary directories."

B.1.4 ERASE

ERASE is a stand-alone maintenance utility used to delete all entries in the temporary screen image library. When an application program has been completed, the temporary library contains the names of all the screen images used. If a new application attempts to access one of those screens, it will receive the temporary copy, perhaps containing unwanted information from the previous run. To eliminate unwanted temporary screen images, the program ERASE is used to delete references to previously used images. Thus, each reference to a new screen image accesses a new, unmodified version of that screen.

ERASE is called with the command:

```
$ ERASE
```

B.1.5 FETCHF

FETCHF is the subroutine which allows a SABERS application program to retrieve user input from a screen image. User input to a single field, any logical group of fields, or the entire screen image may be retrieved with a single call.

A call to FETCHF occurs in the following form:

```
CALL FETCHF (QNAME, BUFFER, BUFSIZ)
```

QNAME represents the qualified name of a screen image, e.g. 'AUTO'; a logical group of fields, e.g. 'AUTO:PRCHINFO'; or a single field, e.g. 'AUTO:PRCHINFO:PRCHDATE:PYEAR'. The QNAME should be a string terminated by a null byte, such as a FORTRAN literal. Screen image and field names must be separated by colons, and the QNAME is enclosed in single quotes.

BUFFER is a series of contiguous bytes of sufficient number to contain the requested information. Values are returned as actual reals, integers and strings; that is, no translation of values from character strings is required of the applications program. When input from a single field is requested, the parameter BUFFER should be a variable of the proper data type and size. However, when input from multiple fields is requested, a contiguous block of bytes is required. This may be accomplished by declaring BUFFER as a byte array of sufficient size, and equivalencing appropriate variables to the BUFFER array; or by placing the appropriate values in a common area to enforce their order and contiguity in storage. In the first case, the parameter BUFFER would be the name of the byte array; in the second case, BUFFER would be the name of the first variable in the common block.

BUFSIZ is an integer variable representing the size in bytes of the variable or block represented by the parameter BUFFER.

As an example, to retrieve the purchase date portion of our AUTO screen image:

```
PARAMETER BUFSIZ = 6

BYTE BUFFER(BUFSIZ)
INTEGER*2 MONTH, DAY, YEAR

EQUIVALENCE (BUFFER(1),MONTH), (BUFFER(3),DAY), (BUFFER(5),YEAR)

CALL FETCHF ('AUTO:PRCHINFO:PRCHDATE', BUFFER, BUFSIZ)
```

The following application-oriented error messages may be generated by FETCHF and are self-explanatory:

"The field specified by the QNAME: ________ does not exist."
"The screen image requested by the QNAME: ________ could not be located in either permanent or temporary directories."
"The field requested by the QNAME: ________ is not a user input field."

B.1.6 NEWFLD

NEWFLD is the subroutine which allows a SABERS applications program to discover which, if any, user input fields in a screen image have been modified by the user after a transaction process. This information can be requested for a single field, a logical group of fields, or the entire screen image with a single call to NEWFLD.

A call to NEWFLD has the following form:

CALL NEWFLD(QNAME,NEW,SIZE).

QNAME represents the qualified name of a screen image, e.g. 'AUTO'; a logical group of fields, e.g. 'AUTO:PRCHINFO'; or a single field, e.g. 'AUTO:PRCHINFO:PRCHDATE:PYEAR'. The QNAME should be a string terminated by a null byte, such as a FORTRAN literal. Screen image and field names must be separated by colons, and the QNAME is enclosed in single quotes.

NEW is an array of Logical*1 variables, one for each user input field requested by QNAME, and, except for single field requests, one for each logical block of fields. That is, a NEWFLD on the screen image AUTO will return 14 logical values: one for each of 11 input fields, one for the entire screen image, one for the logical block PRCHINFO, and one for the logical block PRCHDATE. The array is ordered exactly as the screen image definition file is ordered. The screen image flag is first, followed by succeeding subfields. Thus, the first element in array NEW tells if there has been input to any field in the screen. A logical value .TRUE. indicates new input to the corresponding structure.

SIZE is the number of logical values to be returned in the array NEW.

As an example, to check input to the entire AUTO screen image:

```
LOGICAL*1 NEW(14)

CALL NEWFLD ('AUTO',NEW,14)
```

NEWFLD may generate the following self-explanatory error messages:

" THE FIELD SPECIFIED BY THE QNAME: ______ DOES NOT EXIST. "
" THE SCREEN IMAGE NAME REQUESTED BY THE QNAME: ______ COULD NOT BE LOCATED IN EITHER PERMANENT OR TEMPORARY DIRECTORIES. "
" THE FIELD REQUESTED BY THE QNAME: ______ IS NOT A USER INPUT FIELD. "

B.1.7 PRINTSI

The program PRINTSI allows the SABERS user to generate a hard copy of the most recently transmitted screen image. This is useful in cases where the user has seen a screen image which he was not allowed to edit, and thus could not indicate his desire for a printout. To use, simply type

```
$ PRINTSI
```

and pick up the resulting output.

B.1.8 TXMIT

The program TXMIT is a stand-alone utility program which will allow the applications programmer to inspect a newly created screen image and check it for errors. There are three existing versions of TXMIT designed to take advantage of the characteristics of three different types of terminals: TXMIT40 should be used with the Tektronix 4014 terminal; TXMITVT may be used with the DEC VT-100; and TXMIT16 is designed for the Univac 1652.

To use this utility, simply type:

$ TXMITVT (for example)

The user will be prompted for the name of the screen image to be tested, and a normal transaction process will ensue.

B.1.9 WRONG

WRONG is a special case version of the XMIT routine described below. WRONG is used when an application program has detected an error in user input and that input must be corrected. This is most likely in a case in which the application limits the input options of the user more severely than does the screen image definition.

For example, in our AUTO screen image, the input field PYEAR is constrained to be within the range 1900-2000. A particular application, however, might restrict the range to 1950-2000. If the application inspects user input to this field and finds a value less than 1950, the routine WRONG may be called to request reinput of the value.

A call to WRONG takes the form:

```
CALL WRONG ( QNAME, MSG, REPLY).
```

QNAME is the qualified name of the field which should be reedited, e.g. 'AUTO:PRCHINFO:PRCHDATE:PYEAR', and follows the logical structure of the screen image The QNAME should be a null byte terminated string, such as a FORTRAN literal. Screen image and field names must be separated by colons, and the QNAME is enclosed in single quotes.

MSG is an error message which should be displayed to the user before the screen image is retransmitted. MSG is a byte string which should be terminated by a null byte, such as a FORTRAN literal, but which may not exceed a size of 80 bytes.

REPLY is an error return flag, the values of which are described in full in the documentation on XMIT.

For example:

```
   INTEGER*2 REPLY

   CALL WRONG ('AUTO:PRCHINFO:PRCHDATE:PYEAR',
+'RANGE OF YEARS: 1950 - 2000',
+ REPLY)
```

WRONG may generate the same error messages that are described in the XMIT documentation described below.

B.1.10 XMIT

XMIT is the routine used by an application program to transmit a screen image to the user. Two modes of transmission are allowed: the normal mode in which the user is allowed to make changes to the screen image input fields before returning control to XMIT; and a "no-edit" mode in which the screen is merely displayed to the user and control immediately returns to the application.

In normal mode operation, XMIT checks user input when control has returned to verify that it has conformed to the specifications for each user input field established at screen image creation time. That is, XMIT verifies that input to an integer field is indeed an integer, and that the input integer falls within the specified range of acceptable values. Error checking is not performed for each field immediately on input. Rather, all fields are checked when the user has returned control to the XMIT routine by "sending" the screen. If an error is detected, an error message is generated, the erroneous field is flagged, and the screen is retransmitted to the user. This process continues until all fields are verified.

Certain things can be signalled by the user to the XMIT routine to control phases of subsequent execution. The user can indicate to XMIT that the current screen image should be sent to a hard copy device once input to the screen has been verified and accepted. The user can also request that the editing session be aborted. This causes XMIT to exit, informing the application program of the user's action.

A call to XMIT takes the form:

```
CALL XMIT (SINAME,REPLY)
```

SINAME is the name of the screen image to be transmitted. SINAME should be a string terminated by a null byte, such as a FORTRAN literal.

REPLY is an Integer*2 variable which serves a double function, passing information to the XMIT routine and providing a status return value when the XMIT is completed. When XMIT is called, if the parameter REPLY has the value 0, XMIT proceeds in the normal mode of operations. Any other value for REPLY sets XMIT into "no-edit" mode and causes the specified screen image to be displayed to the user with no opportunity for editing.

Four return values for REPLY are currently defined. On return from XMIT, if the application program discovers that REPLY has a value of 0, XMIT has terminated normally. Non-zero values of REPLY indicate something amiss, and it is suggested that the applications program exit as a result. The value 1 signals that the screen image requested could not be located in the SABERS screen image directories. This would suggest that either the screen image has not yet been created, or that the name has been misspelled. The value 2 indicates that there is a problem in files produced by the CRE8 program. The SABERS system maintainer should probably check this problem. The value 3 signals that the user has requested that the editing session be aborted.

As an example:

```
INTEGER*2 REPLY

REPLY=0
CALL XMIT ('AUTO',REPLY)
IF (REPLY .NE. 0)CALL EXIT
```

XMIT generates the following error messages for the applications programmer:

```
" *** NO SUCH SCREEN IMAGE IN LIBRARY OR LOCAL COPY *** "
" *** ZERO LINKAGE FROM .FLD RECORD TO REQUIRED .SPC *** "
" *** ILLEGAL INPUT FIELD TYPE *** "
```

B.2 TITP REFERENCE MANUAL

The subsections which follow describe in detail the algorithms used in the operation of the SABERS transaction processor. Two general types of routines are included: stand-alone utilities for the creation and maintenance of the SABERS screen image library; and subroutines used in the manipulation of those screen images for user interaction. The routines are presented in alphabetical order as follows:

* B.2.11 TITPIN

* B.2.12 TXMIT

* B.2.13 WRONG

* B.2.14 XMIT

B.2.1 CRE8

The TITP process for presenting a form on a screen for the user to fill in, as part of input to some application, uses the program CRE8 to make the screen images. The application programmer first creates a file containing screen image definition information, then runs CRE8 on that file to compile intermediate files which can be used by the run time TITP programs.

This section describes the structure, algorithms and internal operation of CRE8.

B.2.1.1 The Language

CRE8 is a translator from the screen definition language to internal file forms for direct interpretation by the other TITP routines. The language used is based on the notion of a single rectangular screen area which can be subdivided into nested areas within it called "fields". The whole screen is also considered a field as far as the definition of its characteristics is concerned. Henceforth, the term "field" will mean either a field or a screen image and the term "screen image" will only be used when non-field activities are being discussed.

Each screen image (SI) must have a name since it will occupy a separate set of files which must be referenced by name later. Each field must have a name so that its area within the screen can be read later by TITP.

The characteristics of a field naturally fall into two major features: initial contents definition (or INIT for short) and input specifications.

INIT, for a particular field, consists of a text string which will appear in that field area when TITP presents the screen image to the user. The text string is composed of one or more smaller strings separated by commas. Each smaller string is some number of characters enclosed in matched apostrophies.

The string contents may be characters representing themselves, double slash representing single slash, or a single slash representing the newline symbol or format directives. Further details may be found in the syntax and semantics notes in Section B.2.1.4.

Input specification involves the definition of the form of access to the field which is expected, rules governing the interpretation of input, or the definition of a nested field.

Access determines whether the application program may change the field area and whether the user may change it:

		APPLICATION	
		Input	No-input
USER	Input	FREE	NOA
	No-input	NOU	LOCK

The rules specify the type of input permitted (like "integer", "real"), the requirement for input, if any ("!" meaning mandatory input), and the range of permissible values within the indicated type.

When an internal field is defined, it takes precedence over the input capability of its enclosing field, and therefore, when a field has subfields, it may not have input constraints as well. Subfields have the same definition method as other fields and may, themselves, contain subfields. The purpose of nesting fields in this way is to partition the screen into logically coherent blocks which can later be accessed as a whole.

Initialization is generally not recommended as part of the definition of an input field. The space allocated for the field by its extent is all used for input, so that no text initialized into that space can escape being overwritten by user input. When such label initialization is desired, it is usually best to collect it into the initialization area of the enclosing field.

Additional material describing CRE8 has been presented in Section B.1.1.

B.2.1.2 Relation of SI Elements to File Storage

The file records constructed from a simple screen image are illustrated:

SCREEN DEFINITION:
SINAME: (1:24, 1:80) FREE 'HEADING'
<FIELD: (2:5, 1:10) NOA [I0:999];
SI2: (1:24, 1:20) FREE [A]>

FILE STORAGE:

```
SIDIR.RAN        +SINAME.FLD
 ___________      ______________________________________________________________
|     2     |    |+SINAME | 2 | 0 | 0 | 1 | 24 | 1 | 80 | 3 | 0 | 0 |
| +SINAME   |    |________|___|___|___|___|____|___|____|___|___|___|
| +SI2      |    |+FIELD  | 0 | 0 | 1 | 2 |  5 | 1 | 10 | 2 | 0 | 0 |
|___________|    |________|___|___|___|___|____|___|____|___|___|___|
                      |     |   |   |                     |   |   |_pointer
                      |     |   |   |       extent        |   |___spec length
Directory of          |     |   |   |                     |___access code
Stored screen         |     |   |   |__parent PTR
  images              |     |   |____sibling PTR
                      |     |____child PTR
                      |______name

                  +SI2.FLD
                 _____________________________________________________
                 |+SI2 | 0 | 0 | 0 | 1 | 24 | 1 | 120 | 3 | 0 | 0 |
                 |_____________________________________________________|
                  +SINAME.SPC   empty
                  +SI2.SPC      empty

                  +SINAME.INI                                  1920
                 _____________________________________________________
                 |HEADING                                              |
                 |_____________________________________________________|

                  +SI2.INI                                     2880
                 _____________________________________________________
                 |                                                     |
                 |_____________________________________________________|
```

+Indicates the name immediately following is kept as its hashed value but shown here without hashing for simplicity.

SIDIR.RAN is a permanent file for the user containing a list of the defined screen images.

Each screen image is kept in 3 files. These files are generated by CRE8 and have the names of xxx.FLD, xxx.SPC and xxx.INI, where xxx represents the hashed screen image name. The .FLD file contains a record for each field in the screen image. Each record is structured as shown above. The .SPC file contains all of the additional field specifications for all of the fields in the screen image. The .INI file contains an image of the entire initialization of the screen image.

B.2.1.3 CRE8 Program Details

Calling Map

For clarity, this map does not include the auxiliary routines.

```
CRE8                                      Main Program
  |
  |---SPECS                               Field Specifications Processor
  |     |
  |     |-----FLDNAM                      Field Name Hasher & Allocator
  |     |
  |     |-----EXT                         Field Extent Recognizer
  |     |      |
  |     |      |------TOINT               ASCII to Integer Converter
  |     |
  |     |-----ACESS                       Access Control Word Recognizer
  |     |
  |     |-----STRAIN                      Input Constraint Processor
  |     |       |
  |     |       |-----GETBND              Input Value Bounds Calculator
  |     |
  |     |-----INIT                        Field Content Initialization
  |
  |---SINAME                              Screen Image Name Processor
        |
        |-----LOOKSI                      SI-Name Checker/Allocator
                |
                |-----CONCAT              Append SI File Extents
```

CRE8

CRE8 is the main program of the screen image definition compiling system. It initializes all variables in the central common area COM.COM. It allows entry of the name of the file in which the text of the screen image definition is stored. CRE8 adds the hashed name of the new screen image into the directory file. The screen image definition items are processed into the files described above. CRE8 stops when end-of-file is reached.

The code for the main routine is found in the file CRE8.FLX.

SINAME

This routine is responsible for recognizing the SI name, hashing it and adding the result to the file SIDIR.RAN.

The code for the routine is contained in the file SINAME.FLX.

LOOKSI

If NAME, the input screen image name, is already in SIDIR.RAN, LOOKSI reopens its three files so that subsequent processing will replace its current definition. Otherwise, the SI name is added to SIDIR.RAN and three new files are created.

The code for the program is found in LOOKSI.FLX.

CONCAT

This simply appends an extension EXTN to a name string NAME to produce a valid file name. The calling sequence has the form:

```
CALL CONCAT(NAME,EXTN).
```

The code is contained in the file CONCAT.FTN.

TOINT

This converts an ASCII string representation of a signed integer in STG into that integer in L. The call to TOINT has the form:

```
CALL TOINT(STG,L)
```

The code is found in file TOINT.FLX.

SPECS

SPECS uses an internal stack to control the processing of nested fields and their mutually relative extent bounds. Any field may have a group of subfields. These subfields under a single enclosing field are "siblings." All of the information about the nested structure and the various aspects of the fields and subfields is stored in the .FLD and .SPC files for the current SI, and will be used later by TITP.

The code for the routine is contained in file SPECS.FLX.

FLDNAM

Much like SINAME, FLDNAM recognizes the name of a field, and makes the name a standard size and form using HASH. It then looks the name up in the .FLD file instead of SIDIR.RAN. Name conflicts arise only if the name matches a sibling. These conflicts are fatal to the run. Field Names may be duplicated if they do not share a parent.

The code for the routine is found in the file FLDNAM.FLX.

EXT

EXT recognizes the field extent bound specifications like "(1:3, 25:80)" and puts the actual integer result into the .FLD file. The extent as given in the specification is made into absolute screen coordinates before storing them in the file so that the application need not keep a stack for them.

The code for the routine is contained in the file EXT.NAM.

ACESS

ACESS recognizes one of the access control words "LOCK", "NOU", "NOA", "FREE" and conditions the current field in the screen image initialization accordingly. It sets the high-order bits of protected fields, which cannot receive user inputs. This enables TITP to exclude the screen cursor from these areas.

The code for the routine is found in the file ACESS.FLX.

STRAIN

STRAIN recognizes the various forms of user/application input constraint. "!" is noted, and STRAIN makes the field mandatory if it is present. The input types which is recognizes are:

I = integer
R = real
E = scientific number
S = quoted string

The texts permitted are:

A = alphabetic
D = digit
_ = blanks
$\overline{T}$ = trailing blanks
$ = special characters like @#%^&*({.

The code for the routine is contained in file STRAIN.FLX.

GETBND

GETBND decodes the bounds supplied for input constraints of type I, R, or E. No matter what type was specified, the bounds may be expressed in any of the three forms.

The code for the routine is contained in the file GETBND.FLX.

INIT

Within the confines of the current field, INIT places the initialization string in the proper locations.

The string may contain formatting directives to make them easier to write. The formatting directives currently implemented are described below:

((n) stands for the optional repetition count; 1 is used if n is not present)
NLn New Line
BLn Blanks
'text'n Repeated text pattern

The character '/' is a convenient shorthand for NL or NL1. A quoted blank, ' ', may be used instead of [BL1].

The code for the routine is found in the file INIT.FLX.

COM.COM

This is the common area used throughout CRE8 to communicate the operating variables between routines. The items in COM.COM include record pointers, buffers, file record fields, logical unit numbers, manifest constants, operating limits, access codes, input constraint types, and named ASCII characters. COM.COM contains both declarations of variable type and size, and definitions of all the named common areas, which are:

FILES = File record numbers and counts
UNITS = Logical unit list
FNAMES =Field sections within .FLD file record, associated with offsets
FIXED = Manifest constants to share among routines for regularity
CHARS = The named characters to share among routines for regularity

REPORT

This writes a bracketed message on the terminal to report CRE8 progress. It is called by almost all other CRE8 routines. The routine is called as

```
CALL REPORT(MSG)
```

for the message MSG.

The code is contained in the file REPORT.FTN.

READCH

READCH manages the SI definition input line buffering and strips commentary out of the string.

It includes the entry point for "REVILE", the error condition reporter, so that the current line can be printed on the terminal and the problem spot indicated.

The code for this routine is found in the file READCH.FLX.

INFILE

INFILE is the modularized routine to request the name of a file to access for the screen image definition.

The selected file is opened and attached to the supplied unit number. The routine is called as:

```
    CALL INFILE(UNIT,PARMS,DEFDEV,DEFDIR,DEFNAM,DEFEXT,DEFVRS)
where

    UNIT   = integer logical unit number
    PARMS  = integer number of parameters to follow
    DEFDEV = character default device designator
    DEFDIR = character default directory designator
    DEFNAM = character default filename designator
    DEFEXT = character default filename extent
    DEFVRS = character default filename version number
```

HASH

HASH takes a string name of up to 80 characters and, with the SABERS hashing technique, produces an eight-character alphanumeric file or field name, HSHNAM. Each position in an eight-position table receives a randomized piece of the string. The calling sequence is

```
    CALL HASH(NAME,LENGTH,HSHNAM)
```

where the number of characters in NAME is specified by LENGTH.

The code for the routine is contained in file HASH.FLX.

B.2.1.4 CRE8 Syntax and Semantics

DISPLAY DEFINITION LANGUAGE

```
<SPEC>          ::=<SCREEN IMAGE> [;<SCREEN IMAGE> +]
<SCREEN IMAGE>::=<FIELD>
<FIELD>         ::=[<NAME>]:<EXTENT> [<ACCESS>] [<INIT>] [<RULES> ! <FIELDLIST>]
<NAME>          ::= <CHAR STRING EXCEPT COLON OR BLANK>
<FIELDLIST>     ::= "<" <FIELD> [; <FIELD> +] ">"
<EXTENT>        ::=( <RANGE> , <RANGE> )
<RANGE>         ::= <POSINT> : <POSINT>
<ACCESS>        ::= FREE ! NOU ! NOA ! LOCK
<INIT>          ::= '<INITSTRING>' [,'<INITSTRING>' +]
<INITSTRING>    ::=<INIT ITEM> ! <INITSTRING><INIT ITEM>
<INIT ITEM>     ::=<CHAR ITEM> ! <DOUBLE SLASH> ! <FORMAT DIRECTIVES> !<NEWLINE>
<CHAR ITEM>     ::=<CHAR> ! <DOUBLE QUOTE>
<DOUBLE QUOTE>::= ''
<DOUBLE SLASH>::=//
<NEWLINE>       ::=
<FORMAT DIRECTIVES>::="[" <DIRS> "]"
<DIRS>          ::=<DIR> [,<DIR>]
<DIR>           ::= NL[<COUNT>] ! BL[<COUNT>] ! '<STRING>'[<COUNT>]
<COUNT>         ::=<POSINT>
<RULES>         ::= "[" <INPUT SPECS> "]"
<INPUT SPECS> ::= ["!"] <SPECL>
<SPECL>         ::= I [<BOUND>] ! R [<BOUND>] ! E [<BOUND>] !
                    S'<STRING>' [,'<STRING>'] ! [A] [D] [_] [$] [T] ['<STRING>']
<BOUND>         ::= <EFORM>:<EFORM>
<EFORM>         ::= <INT> [.<POSINT>] [E <INT>]
<INT>           ::= <POSINT> ! +<POSINT> ! -<POSINT>
<POSINT>        ::= <DIGIT> ! <POSINT><DIGIT>
<STRING>        ::= <CHAR ITEM> ! <STRING><CHAR ITEM>
```

NOTES:

THE DIGRAPH "+]" MEANS ZERO OR MORE TIMES.
BRACKETS [] SURROUND OPTIONAL INCLUSIONS.
QUOTES "" SURROUND TERMINAL SYMBOLS WHICH MIGHT BE CONFUSED WITH METASYMBOLS.
THE EXCLAMATION POINT ! IS USED HERE TO MEAN "OR".

SEMANTICS:
1) INPUT IS ALLOWED ONLY ON NON-LOCKED FIELDS.
2) EXTENT VALUES ARE CONSTRAINED BY THE SIZE IMPLICIT IN THE ENCLOSING OR NEXT OUTER FIELD.
3) INPUT IS DEFINED ONLY FOR INNERMOST FIELDS.
4) MANDATORY INPUT, SIGNIFIED BY "!", IS MEANINGFUL ONLY FOR "FREE" OR "NOA"
5) NO FIELD NESTING DEPTH LIMIT IS DEFINED.
6) MAXIMUM INTEGER MAGNITUDE IS 32767.
7) A STRING, BEGINNING WITH THE DIGRAPH "(*" AND CONTAINING ANY COMBINATION OF CHARACTERS EXCEPT THE DIGRAPH "*)", IS CONSIDERED COMMENTARY AND IS COMPLETELY IGNORED IN PROCESSING A DISPLAY DEFINITION NO MATTER WHERE SUCH A COMMENT OCCURS.
8) FIELD EXTENTS MAY NOT OVERLAP.
9) WITHIN INIT STRINGS A DOUBLE QUOTE YIELDS A QUOTE CHARACTER IN STORAGE, A DOUBLE SLASH YIELDS A SLASH CHARACTER IN STORAGE. TO GET DOUBLE NEWLINE, SLASHES MUST BE SEPARATED BY AT LEAST ONE OTHER CHARACTER, BUT THE FORM [NL2] IS PREFERRED.
10) IN FORMAT DIRECTIVES, 1 IS THE DEFAULT COUNT.

SPECIAL SYMBOLS:

: END OF FIELD NAME,
RANGE SEPARATOR,
BOUND SEPARATOR

() ENCLOSE AN EXTENT,
MAY BE INVOLVED WITH A COMMENT

[] ENCLOSE INPUT SPECIFICATIONS,
ENCLOSE FORMAT DIRECTIVES IN INIT STRING

<> ENCLOSE SUBFIELD LIST

(* *) BEGINNING AND END OF COMMENTARY

, SEPARATE STRINGS IN "S" INPUT SPEC,
MAY SEPARATE INIT STRINGS

_	(UNDERSCORE) "BLANKS ALLOWED" TEXT INPUT SPEC
A	"ALPHABET ALLOWED" TEXT INPUT SPEC
D	"DIGIT ALLOWED" INPUT SPEC
T	"TRAILING BLANKS ALLOWED" INPUT SPEC
$	"SPECIAL CHARS ALLOWED" INPUT SPEC
/	NEWLINE SYMBOL IN INIT STRING
//	SLASH ("/") SYMBOL IN INIT STRING
+-	SIGN FOR IMMEDIATELY FOLLOWING NUMBER
!	PREFIX TO INPUT SPEC INDICATING MANDATORY USER INPUT

NOTES:
BLANKS, LINE BOUNDARIES, TABS ETC. ARE SIGNIFICANT ONLY IN QUOTED STRINGS. BLANKS WITHIN FORMAT DIRECTIVES IN INIT STRINGS ARE IGNORED.

B.2.2 DELSI

The routine DELSI is a stand-alone utility which allows the user to delete a Screen Image from the permanent SABERS Screen Image directory.

The user is prompted for the name of the Screen Image to be deleted. After reading in the name, the subroutine HASH is called to hash this name into a unique eight-character identifier. This identifier will be used in the search through the SABERS screen image library. The library file 'SIDIR.RAN' is opened and the first record read in. Each record in this file is six words long. In all but the first record, the first four words represent an eight-character hashed name of an existing screen image. In record 1, word 5 represents the total number of records (that is, the number of screen images + 1) represented in the library. DELSI then reads each succeeding record until either the hashed name matches the requested name, or end-of-file is reached. If end-of-file is reached, this indicates that the requested screen image did not exist in the library and the user is so informed. Otherwise, the located record is deleted from the file and the file is compacted. The user is then provided with the hashed name so that the .FLD, .SPC and .INI files may be deleted manually.

The code for DELSI is to be found in the file DELSI.FLX .

B.2.3 DM

The routine DM is the terminal-specific device manager for TITP. It is through DM that the routines XMIT and WRONG interface to the SABERS user. This routine is not application program callable; it is strictly internal to TITP itself.

DM can perform three distinct operations related to user I/O on the terminal. It can perform a simple clear of the screen; it can produce a single line of output; and it can perform all input and output operations necessary to interface with the user in a complete transaction process.

The call to DM takes the following form:

```
CALL DM ( ID, TRTYP, DEVTYP, MSG )
```

ID is the four-character identification of the calling program.

TRTYP is the four-character identification of the desired transaction type. Legal values are: "TITP" for a transaction process; "TIGP" for a graphics process; and "TIST" for serial text.

DEVTYP is the four-character identifier of a target terminal. For example, "TEKT" might indicate a Tektronix 4014.

MSG is the text of the message to be displayed. MSG is a byte array with leading integers describing its shape and size. The first integer represents the dimensionality of the message. Generally, single line output would be two-dimensional (having rows and columns). A transaction process could be two or three dimensional, the third dimension indicating multiple pages. Following the dimension specification is a series of numbers describing the size of each dimension in the following order: rows, columns, pages. A two-dimensional message will not have a value for pages. Lastly, there is an

integer representing the total length of the message in bytes, including the leading integers.

When DM is called, it first decides which type of transaction is being requested. For a clear-screen request, this action is performed and control immediately returns. For a single line output request, the size of the message is calculated; the screen is cleared; and the message is written out to the user.

For a full-scale transaction process, the message is written out in the specified row and column format for the user. Two character conventions are used to flag certain conditions for DM. A screen image generally will have areas within it (fields) into which the user may insert data. Other areas will be locked from user editing. DM uses the sign bit of each byte in the screen image to determine whether or not the user may make changes there. A byte with the sign bit on is a no-user byte and the cursor is never positioned there. A byte with the sign bit off is editable by the user, and the cursor may be positioned there if the user so requests.

The second character convention is the treatment of question marks. When the caller wants to tell the SABERS user that an input field has been filled with an improper response, the offending field is filled with question marks and the screen image is redisplayed to the user. Certain implementations of DM (depending on terminal type) allow the field to be further highlighted by putting the question marks in reverse video and blinking them. Thus the user can more readily identify the problem area.

After display of the screen image, a flag is checked to determine if the user should be allowed to edit the screen. If not, control returns. If so, the first input field in the screen is located and the cursor is moved to that position. Input is then accepted from the user. He has many options. He may type in characters as input to the field. He may use an extensive set of cursor movement commands to reposition the cursor to a desired input field.

He may request that the screen image be printed after validation. He may ask that the editing session be aborted. Instructions may be displayed (each terminal type will have its own instuction set). Finally he may send the screen back to the caller for processing.

Input to the device manager is accepted on a character-by-character basis, and actions are performed based on those characters. Basically, any non-control characters are inserted directly into the screen image buffer at the current cursor position. Editing and other commands consist of escape sequences, that is, a special escape character followed by one or more additional characters. Parsing these escape sequences permits DM to follow the user commands.

Once the user signals his desire to send the screen for processing, DM returns MSG updated with the user input.

The code for the device managers may be found in the following files: DMTPVT.FLX for the VT-100; DMTP16.FLX for the UNIVAC 1652; and DMTP40.FLX for the TEKTRONIX 4014.

B.2.4 EDIT

EDIT is the subroutine which allows a SABERS application to insert values into a screen image prior to displaying the screen to the user. The routine permits editing of a single field, any logical group of fields, or the entire screen image with a single call.

EDIT calls the HASH routine to get the hashed version of the screen image name portion of the parameter QNAME. The temporary screen image directory, 'SIDIR.TMP', is then searched to see if the screen image is already in use. If it is not found, the routine FETCH is called to access the permanent copy of the screen image. If there is no such screen image, an error message is generated. Once the screen image is located, the .FLD and .TEM files are opened. The current screen image template is read in from the .TEM file.

A search through the screen image's logical structure, by way of the .FLD file is performed. Each field in the logical block is edited according to its data type. Any field flagged as 'no-application' or 'locked' access is simply skipped and editing continues with the next field in the block.

When editing is completed the modified screen image is written back to the .TEM file for later access by other routines. Control is then returned to the calling program.

It should be noted that EDIT performs no type or range checking. The appropriate number of bytes (e.g. four for a REAL) are translated and the resulting characters inserted into the allotted number of bytes in the screen image. The operating system will generate format conversion errors if an improper conversion is attempted.

The code for EDIT is to be found as an entry point to the subroutine FETCHF in the file FETCHF.FLX .

NOTE: due to restrictions imposed by the compatibility mode implementation of SABERS (specifically, the time required to pass a large variable in a subroutine call), communication of the parameter BUFFER is accomplished through file I/O (transparently to the application program). For this reason, the EDIT entry point has only one parameter, QNAME.

B.2.5 ERASE

ERASE is a stand-alone maintenance utility used to delete all entries in the temporary screen image library. When an application program has terminated, the temporary library 'SIDIR.TMP' contains names of all screen images accessed. If a new application attempts to access one of these screens, it will receive the temporary copy, perhaps containing unwanted information from the previous run. In order to provide the user with an unmodified screen image, the program ERASE is used to delete references to previously used screen images. Thus, each reference to a new screen image accesses a new version of that screen.

ERASE opens the temporary library, and the first record is read in to determine the number of entries in the file. That number is reset to 1 (the only entry will be the record count) and then all previously used records are deleted by writing over them. The program then exits.

The code for the ERASE utility is found in the file ERASE.FLX .

B.2.6 FETCH

FETCH is a subroutine for internal TITP use only. It should not be called by applications programs. FETCH is used to retrieve a copy of a screen image from the permanent directory and establish a temporary, editable version.

FETCH opens the permanent screen image directory ('SIDIR.RAN') and compares each hashed name contained therein to the hashed name argument. If a match is found, the appropriate .INI file is opened, its contents read, and those contents written back out to a .TEM file. If no match is found for the hashed name, an error flag is returned.

The code for FETCH is located in the file FETCH.FLX .

B.2.7 FETCHF

FETCHF is the subroutine which allows a SABERS applications program to retrieve user input from a screen image. User input to a single field, any logical group of fields, or the entire screen image may be retrieved with a single call.

FETCHF calls the routine HASH to produce the unique eight-character hashed version of the screen image name. The temporary screen image directory ('SIDIR.TMP') is searched to see if the screen image is already in use. If it is not found, the routine FETCH is called to access the permanent copy of the screen image. Once the screen image is located the .FLD file is opened.

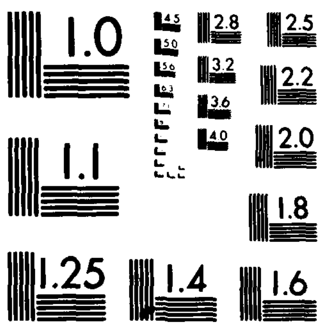
A search through the screen image's logical structure by way of the .FLD file is performed. Input to each user-editable field within the logical block is retrieved according to its data type. Any field flagged as 'locked' or 'no-user' is simply skipped and input retrieval continues with the next field in the block.

The code for FETCHF can be found in the file FETCHF.FLX. The VAX compatability mode environment has imposed several constraints on the FETCHF routine which will be eliminated in native mode. The transfer of the retrieved input data between FETCHF and the application is actually accomplished through file I/O rather than sending interprocess messages. Only the name of the screen image (parameter QNAME) is passed to the FETCHF routine itself, and the only return value is a logical flag signalling if the entire screen image is being requested. This flag tells the communications interface on the application side whether to retrieve the requested data from the first or the second record in the file "FETCHF.DAT." The first record contains all user input information from the screen image. The second record is the subset of the user input if retrieval of a field or group of fields was requested. The communications interface requires the length of this buffer (from the parameter LENGTH) to perform the file read. When SABERS is placed in native

AD-A110 272 PAR TECHNOLOGY CORP ROME NY F/G 9/2
SABERS. STAND-ALONE ADIC BINARY EXPLOITATION RESOURCES SYSTEM. --ETC(U)
SEP 81 A J FRANKLIN, R L CALDWELL, S COLE F30602-78-C-0078
UNCLASSIFIED RADC-TR-81-250-VOL-2 NL

4 of 4

END
DATE FILMED
2-82
DTIC



MICROCOPY RESOLUTION TEST CHART

mode, both the subroutine and a reference to it should use the parameters (QNAME,BUFFER).

B.2.8 HASH

HASH is a routine designed to take a name of arbitrary length and convert it to a unique eight-character hashed name. Hashed names are constructed from letters and digits giving 36 possible building blocks and nearly three trillion unique names. However, this does not entirely preclude the possiblity of collisions, so it may occur that two screen image names could hash to the same names. Should this happen, one screen image name must be changed.

The HASH routine is not callable by the application program but is an internal TITP routine. HASH receives as parameters the name to be hashed, its length, and the address of an array in which to store the hashed name. The hashing function is applied to each of the characters in the unhashed name in turn until all are processed. The function then returns to the calling program.

The code for the HASH routine is located in file HASH.FLX .

B.2.9 NEWFLD

NEWFLD is the subroutine which allows a SABERS application program to discover which, if any, user input fields in a screen image have been modified by the user after a transaction process. This information can be requested for a single field, a logical group of fields or the entire screen image with a single call to NEWFLD.

NEWFLD shares much of its code with the routines EDIT and FETCHF which have similar functional requirements. The HASH routine is called to produce the unique eight-character hashed version of the screen image name. The temporary screen image directory is searched to see if the screen image is already in use. If it is not found, the routine FETCH is called to access the permanent copy of the screen image. Once the screen image is located a search is performed to locate the logical block specified by QNAME. The values stored by XMIT in the NEWFLD.DAT file which correspond to the located fields are sorted out and returned to the application program.

The code for the routine NEWFLD is located as an entry point to the routine FETCHF in the file FETCHF.FLX. The SABERS environment, using compatability mode on the VAX, has constrained the means by which the NEW array is passed back to the calling program. Rather than pass a possibly very large array through the communications interface, it was decided to use file I/O, transparent to the applications. When SABERS becomes a native mode system, this constraint will be removed and the array may be passed using the conventional parameter passing method in FORTRAN. The parameter SIZE can then be omitted.

The code for the routine NEWFLD is located as an entry point to the routine FETCHF in the file FETCHF.FLX .

B.2.10 PRINTSI

The program PRINTSI allows the SABERS user to generate a hard copy of the most recently transmitted screen image. This is useful in cases where the user has seen a screen image which he was not allowed to edit, and thus could not indicate his desire for a printout.

PRINTSI finds the name of the screen image to be printed in a file called HASHNAME.DAT. The extent '.TEM' is appended to the name and the file with that name is read in and sent to the hard copy device.

B.2.11 TITPIN

TITPIN was created to allow for convenient initialization of the TITP process at startup. Currently it is an empty program which simply returns control immediately. It may be dispensed with when SABERS is put into VAX native mode.

The code for TITPIN is located in the file TITPIN.FLX .

B.2.12 WRONG

WRONG is an entry point in the XMIT routine described below, which is used when an application program has detected an error in user input and must direct the user to correct the input. This is most likely in a case in which the application limits the input options of the user more severely than is provided for in the screen image definition.

For example, in a screen image, the input field YEAR may be constrained to be within the range 1900-2000. A particular application, however, might restrict the range further to 1950-2000. If the application inspects user input to this field and finds a value less than 1950, the routine WRONG may be called to request reinput of the value.

WRONG is located in the file XMIT.FLX .

B.2.13 XMIT

XMIT is the routine used by an application program to transmit a screen image to the user. Two modes of transmission are allowed: the normal mode in which the user is allowed to make changes to the screen image input fields before returning control to XMIT; and a "no-edit" mode in which the screen is merely displayed to the user and control immediately returns to the application.

In normal mode operation, XMIT checks the user input to verify that it conforms to the specifications for each user input field established at screen image creation time. That is, XMIT verifies that input to an integer field is indeed an integer, and that the input integer falls within the specified range of acceptable values. Error checking is not performed for each field immediately on input. Rather, all fields are checked when the user has returned control to the XMIT routine by transmitting the screen. If an error is detected, an error message is generated, the erroneous field is flagged, and the screen is retransmitted to the user. This process continues until all fields are verified.

Certain information can be signalled by the user to the XMIT routine to control phases of subsequent execution. The user can indicate to XMIT that the current screen image should be sent to a hard copy device once input to the screen has been verified and accepted. The user can also request that the editing session be aborted. This causes XMIT to exit, informing the application program of the user's action.

XMIT generates the following error messages for the applications programmer:

```
" *** NO SUCH SCREEN IMAGE IN LIBRARY OR LOCAL COPY *** "
" *** ZERO LINKAGE FROM .FLD RECORD TO REQUIRED .SPC *** "
" *** ILLEGAL INPUT FIELD TYPE *** "
```

When XMIT is called, the routine performs a search through the temporary screen image directory to see if the named screen image is currently in use. If not, an attempt is made to FETCH the image from the permanent directory. An unsuccessful FETCH causes an error condition to occur and the appropriate error message and value for the return parameter REPLY are generated. The hashed screen image name is placed in a file called HASHNAME.DAT for possible later use by the PRINTSI utility.

The various screen image files (.FLD,.TEM,.SPC) are opened and the temporary copy of the screen image is read in. A copy of the screen is made and kept in core so that after user input a comparison can be made to determine which fields have received new data. The screen is transmitted to the user by way of the routine DM which handles all the terminal-specific details of user I/O. A flag signals DM if XMIT is in "no-edit" mode and DM returns control immediately following the display of the screen image to the user. XMIT subsequently returns to the application program.

When an editable screen image has returned from the user, the KILL flag is examined to determine if the user has requested termination of the editing session. If so, XMIT returns to the application with a REPLY value of 3. Otherwise, the user copy of the screen image is stepped through field by field. Comparison is made to the core copy of the screen image to determine if the user has changed the contents of the field. If so, the user input is then validated according to the field type and specifications. When faulty input is detected, the erroneous field is filled with question marks and the screen is retransmitted to the user for another round of editing. Full verification of each field is performed each time the screen image returns from the user in case he has made changes to fields other than the erroneous one.

Once the screen has been verified, several closing operations are performed. The verified screen image is written back out to the .TEM file and all currently open files are closed. An array of user input flags is written

out to a file NEWFLD.DAT for later use by the routine NEWFLD. All values contained in user input fields are collected and written to the file FETCHF.DAT for later retrieval by the FETCHF routine. If the user has requested that the verified screen image be printed, the screen image is written to the hard copy device. XMIT may then return control to the applications program.